FIG.1

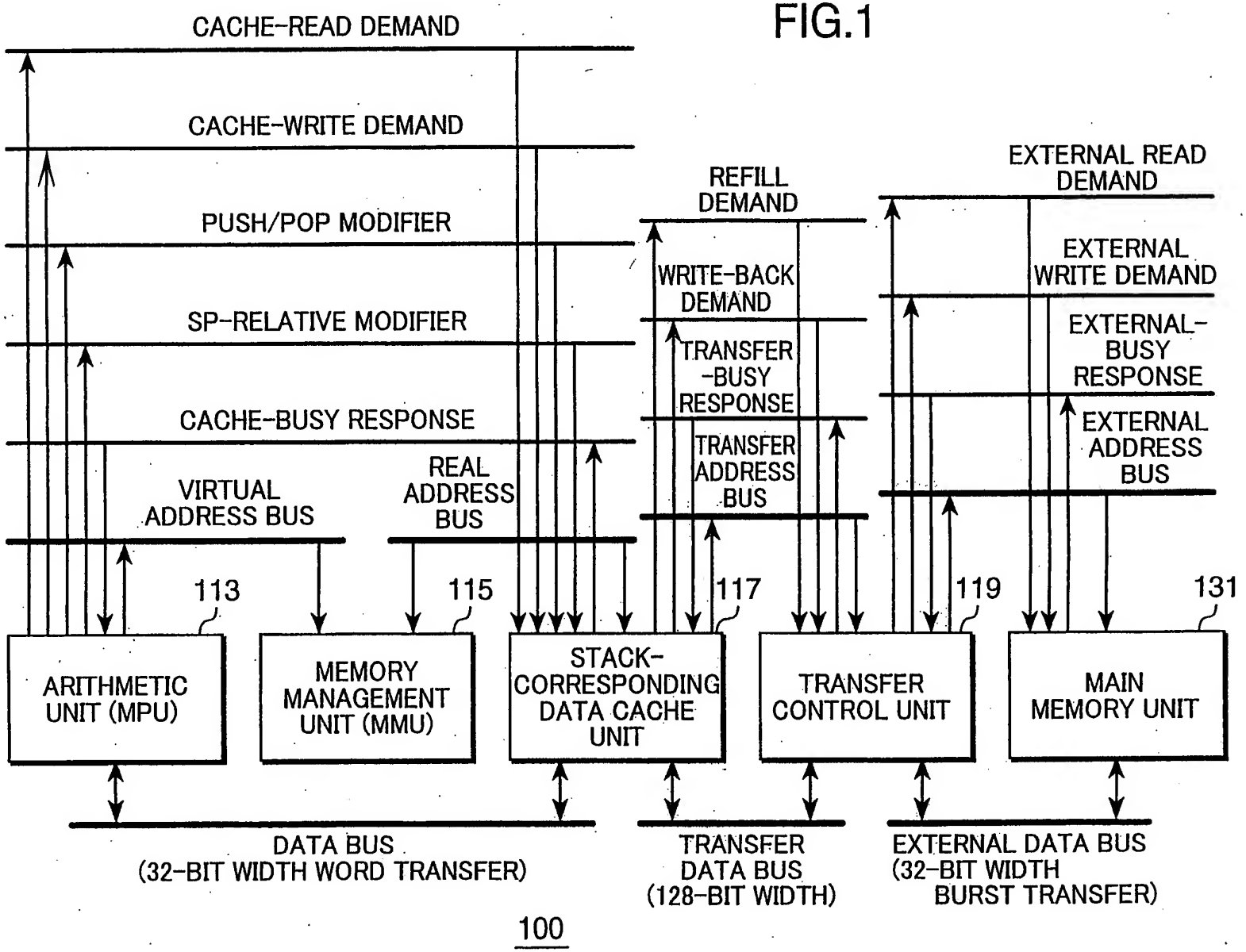

CACHE-READ DEMAND
CACHE-WRITE DEMAND
PUSH/POP MODIFIER
SP-RELATIVE MODIFIER
CACHE-BUSY RESPONSE
VIRTUAL ADDRESS BUS
REAL ADDRESS BUS
REFILL DEMAND
WRITE-BACK DEMAND
TRANSFER -BUSY RESPONSE
TRANSFER ADDRESS BUS
EXTERNAL READ DEMAND
EXTERNAL WRITE DEMAND
EXTERNAL-BUSY RESPONSE
EXTERNAL ADDRESS BUS
113
115
117
119
131
ARITHMETIC UNIT (MPU)
MEMORY MANAGEMENT UNIT (MMU)
STACK-CORRESPONDING DATA CACHE UNIT
TRANSFER CONTROL UNIT
MAIN MEMORY UNIT
DATA BUS (32-BIT WIDTH WORD TRANSFER)
TRANSFER DATA BUS (128-BIT WIDTH)
EXTERNAL DATA BUS (32-BIT WIDTH BURST TRANSFER)
100

FIG.2

	INSTRUCTION	OPERATION
(a)	PUSH Rn,[−Rm]	Rm=Rm−4 [Rm]=Rn PUSH/POP MODIFIER=1
(b)	POP [Rn+],Rm	PUSH/POP MODIFIER=1 Rm=[Rn] Rn=Rn+4
(c)	LOAD [Rn+OFFSET],Rm	SP-RELATIVE MODIFIER=1 Rm=[Rn+OFFSET]
(d)	STORE Rn,[Rm+OFFSET]	[Rm+OFFSET]=Rn SP-RELATIVE MODIFIER=1

FIG.3

	CACHE-READ DEMAND	CACHE-WRITE DEMAND	PUSH/POP MODIFIER	SP-RELATIVE MODIFIER	INSTRUCTION/ OPERATION
(a)	0	0	×	×	—
(b)	0	1	0	0	RANDOM WRITE
(c)	0	1	0	1	SP-RELATIVE WRITE
(d)	0	1	1	0	PUSH
(e)	0	1	1	1	—
(f)	1	0	0	0	RANDOM READ
(g)	1	0	0	1	SP-RELATIVE READ
(h)	1	0	1	0	POP
(i)	1	0	1	1	—
(j)	1	1	×	×	—

FIG.4

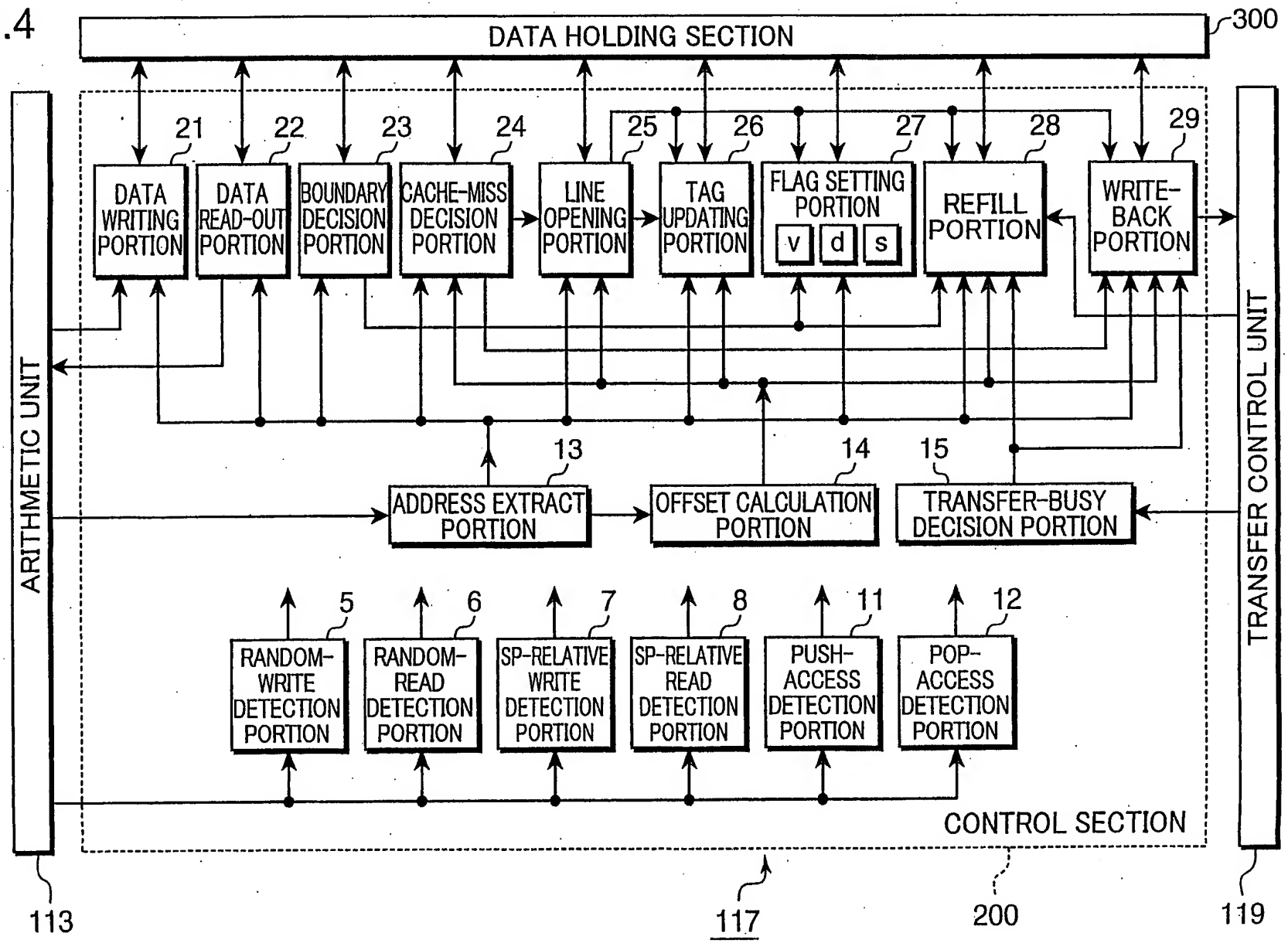

DATA HOLDING SECTION
300
21
22
23
24
25
26
27
28
29
DATA WRITING PORTION
DATA READ-OUT PORTION
BOUNDARY DECISION PORTION
CACHE-MISS DECISION PORTION
LINE OPENING PORTION
TAG UPDATING PORTION
FLAG SETTING PORTION
v
d
s
REFILL PORTION
WRITE-BACK PORTION
ARITHMETIC UNIT
TRANSFER CONTROL UNIT
13
14
15
ADDRESS EXTRACT PORTION
OFFSET CALCULATION PORTION
TRANSFER-BUSY DECISION PORTION
5
6
7
8
11
12
RANDOM-WRITE DETECTION PORTION
RANDOM-READ DETECTION PORTION
SP-RELATIVE WRITE DETECTION PORTION
SP-RELATIVE READ DETECTION PORTION
PUSH-ACCESS DETECTION PORTION
POP-ACCESS DETECTION PORTION
CONTROL SECTION
113
117
200
119

FIG.5

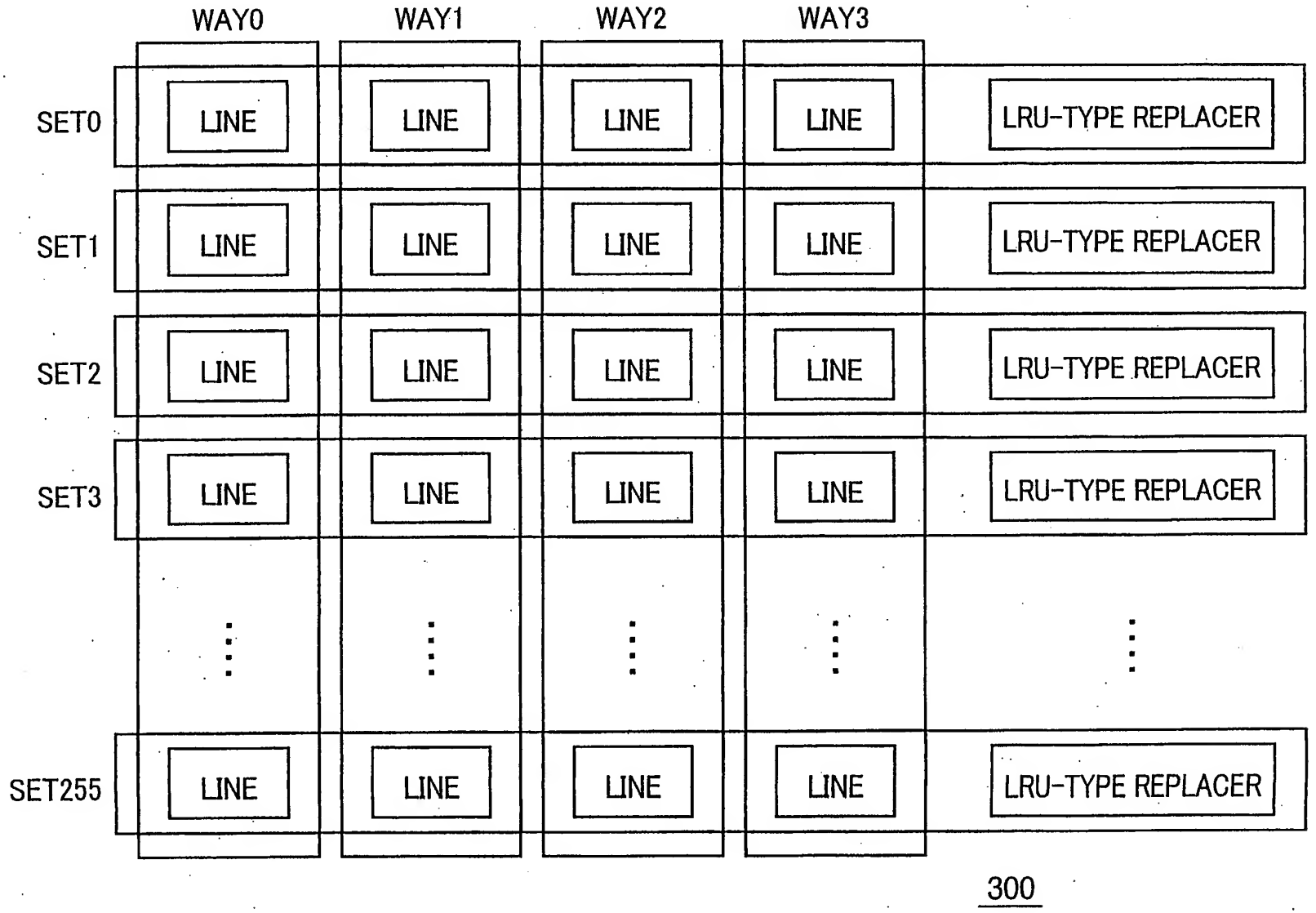

WAY0
WAY1
WAY2
WAY3
SET0
SET1
SET2
SET3
SET255
LINE
LRU-TYPE REPLACER
300

FIG.6

	TAG	VALID FLAG	DIRTY FLAG	STACK FLAG	DATA AREA
LINE1	20 BITS	1 BIT	1 BIT	1 BIT	16 BYTES(4 WORDS)
LINE2	20 BITS	1 BIT	1 BIT	1 BIT	16 BYTES(4 WORDS)
LINE3	20 BITS	1 BIT	1 BIT	1 BIT	16 BYTES(4 WORDS)
LINE4	20 BITS	1 BIT	1 BIT	1 BIT	16 BYTES(4 WORDS)
⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮

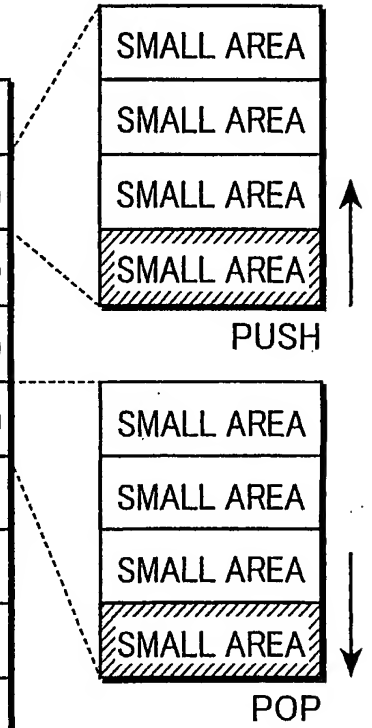


FIG.7

	LINE STATE			MEANING
	VALID FLAG	DIRTY FLAG	STACK FLAG	
(a)	0	0	0	THE CONTENTS OF THE MEMORY INDICATED BY THE TAG ARE NEWER THAN THE CONTENTS OF THE LINE (THIS LINE IS NOW OPENED)
(b)	0	0	1	THE LINE WAS ALLOCATED TO THE STACK. THE CONTENTS OF THE MEMORY INDICATED BY THE TAG ARE NEWER THAN THE CONTENTS OF THE LINE (THIS LINE IS NOW OPENED)
(c)	0	1	0	—
(d)	0	1	1	—
(e)	1	0	0	THE CONTENTS OF THE MEMORY INDICATED BY THE TAG COINCIDE WITH THE CONTENTS OF THE LINE
(f)	1	0	1	THE LINE IS ALLOCATED TO THE STACK. THE CONTENTS OF THE MEMORY INDICATED BY THE TAG COINCIDE WITH THE CONTENTS OF THE LINE
(g)	1	1	0	THE CONTENTS OF THE LINE ARE NEWER THAN THE CONTENTS OF THE MEMORY INDICATED BY THE TAG
(h)	1	1	1	THE LINE IS ALLOCATED TO THE STACK. THE CONTENTS OF THE LINE ARE NEWER THAN THE CONTENTS OF THE MEMORY INDICATED BY THE TAG

FIG.8

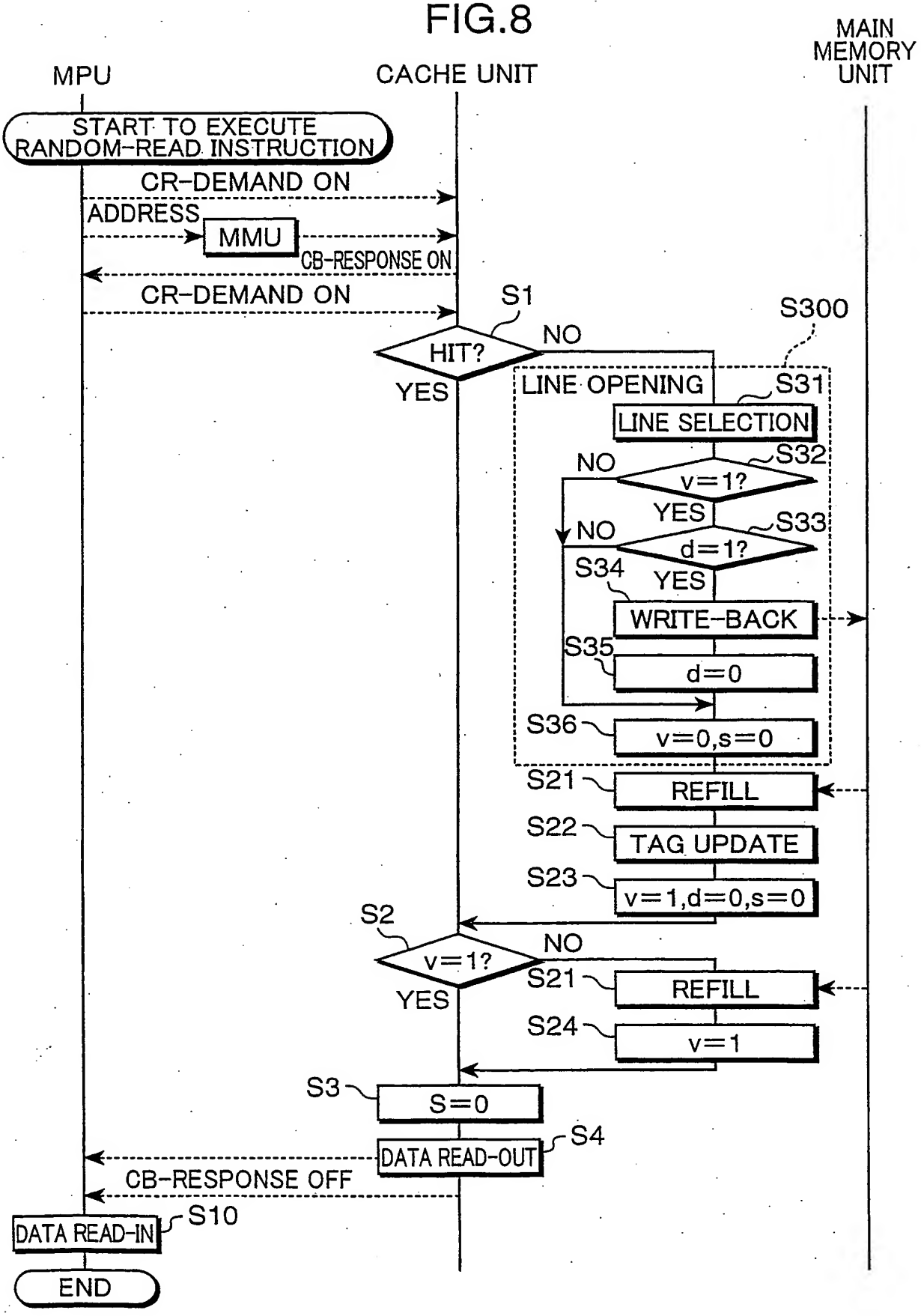

MPU
CACHE UNIT
MAIN MEMORY UNIT
START TO EXECUTE RANDOM-READ INSTRUCTION
CR-DEMAND ON
ADDRESS
MMU
CB-RESPONSE ON
CR-DEMAND ON
S1
HIT?
NO
YES
S300
LINE OPENING
S31
LINE SELECTION
S32
NO
v=1?
YES
S33
NO
d=1?
S34
YES
WRITE-BACK
S35
d=0
S36
v=0,s=0
S21
REFILL
S22
TAG UPDATE
S23
v=1,d=0,s=0
S2
v=1?
NO
YES
S21
REFILL
S24
v=1
S3
S=0
S4
DATA READ-OUT
CB-RESPONSE OFF
S10
DATA READ-IN
END

FIG.9

	LINE STATE			OPERATION	
	VALID FLAG	DIRTY FLAG	STACK FLAG	RANDOM READ	RANDOM WRITE
(a)	NOT ALLOCATED			DECISION OF THE LINE TO BE OPENED, OPENING OF THE DECIDED LINE, REFILL, OPERATION OF (b)	DECISION OF THE LINE TO BE OPENED, OPENING OF THE DECIDED LINE, REFILL, OPERATION OF (b)
(b)	1	0	0	DATA READ	DATA WRITE d=1
(c)	1	0	1	DATA READ s=0	DATA WRITE d=1 s=0
(d)	1	1	0	DATA READ	DATA WRITE
(e)	1	1	1	DATA READ s=0	DATA WRITE s=0
(f)	0	0	0	REFILL v=1 DATA READ	REFILL v=1 DATA WRITE d=1
(g)	0	0	1	REFILL v=1 DATA READ s=0	REFILL v=1 DATA WRITE d=1 s=0

FIG.10

	LINE STATE			OPERATION	
	VALID FLAG	DIRTY FLAG	STACK FLAG	SP MOVES UP (SP−TAG≧4WORDS)	OPENING OF THE LINE
(a)	NOT ALLOCATED				—
(b)	1	0	0		v=0
(c)	1	0	1		v=0 s=0
(d)	1	1	0		WRITE-BACK v=0 d=0
(e)	1	1	1	WRITE-BACK d=0	WRITE-BACK v=0 d=0 s=0
(f)	0	0	0		
(g)	0	0	1		s=0

FIG.11

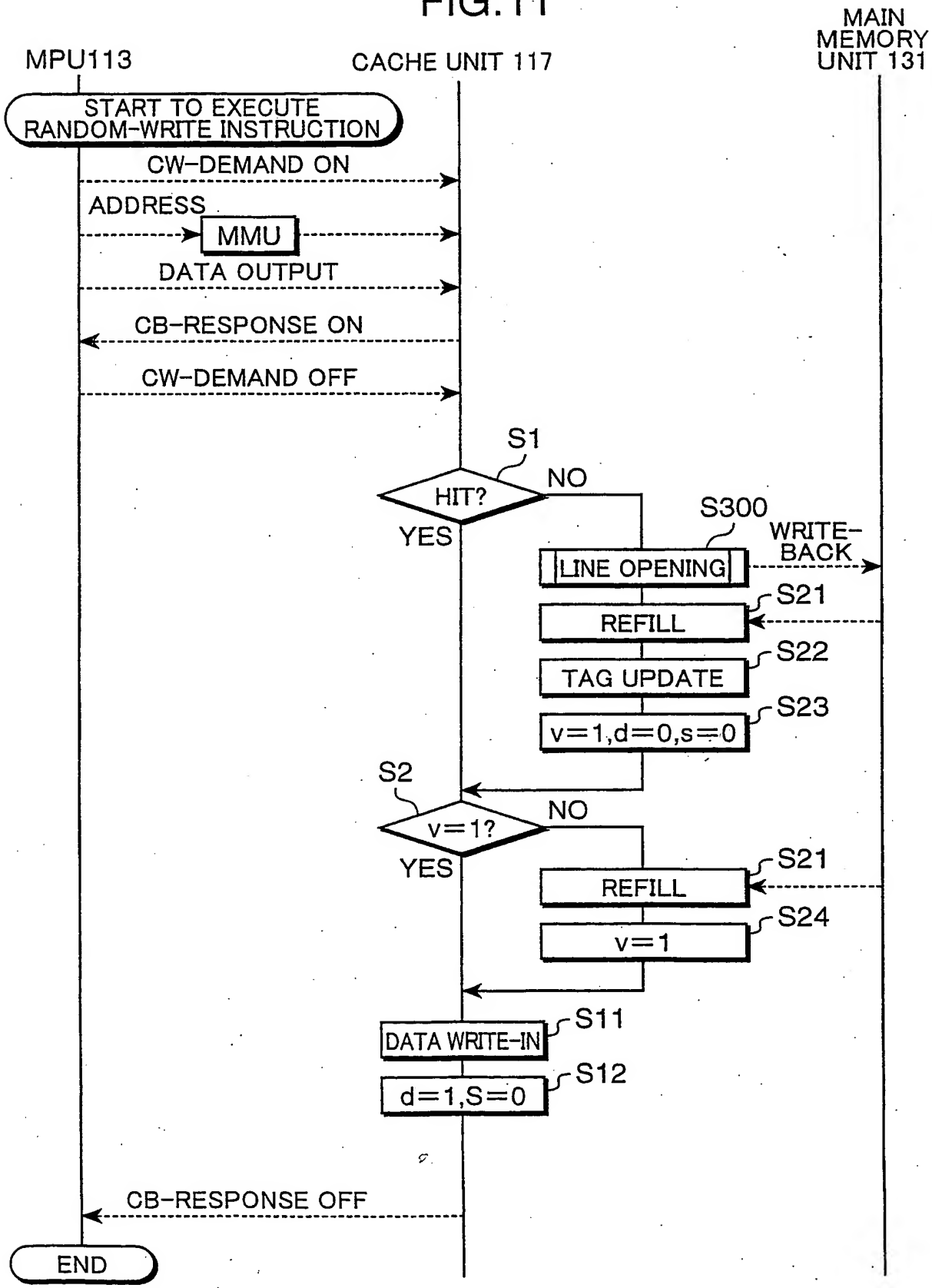

MPU113
CACHE UNIT 117
MAIN MEMORY UNIT 131
START TO EXECUTE RANDOM-WRITE INSTRUCTION
CW-DEMAND ON
ADDRESS
MMU
DATA OUTPUT
CB-RESPONSE ON
CW-DEMAND OFF
S1
HIT?
NO
YES
S300
WRITE-BACK
LINE OPENING
S21
REFILL
S22
TAG UPDATE
S23
v=1,d=0,s=0
S2
v=1?
NO
YES
S21
REFILL
S24
v=1
S11
DATA WRITE-IN
S12
d=1,S=0
CB-RESPONSE OFF
END

FIG.12

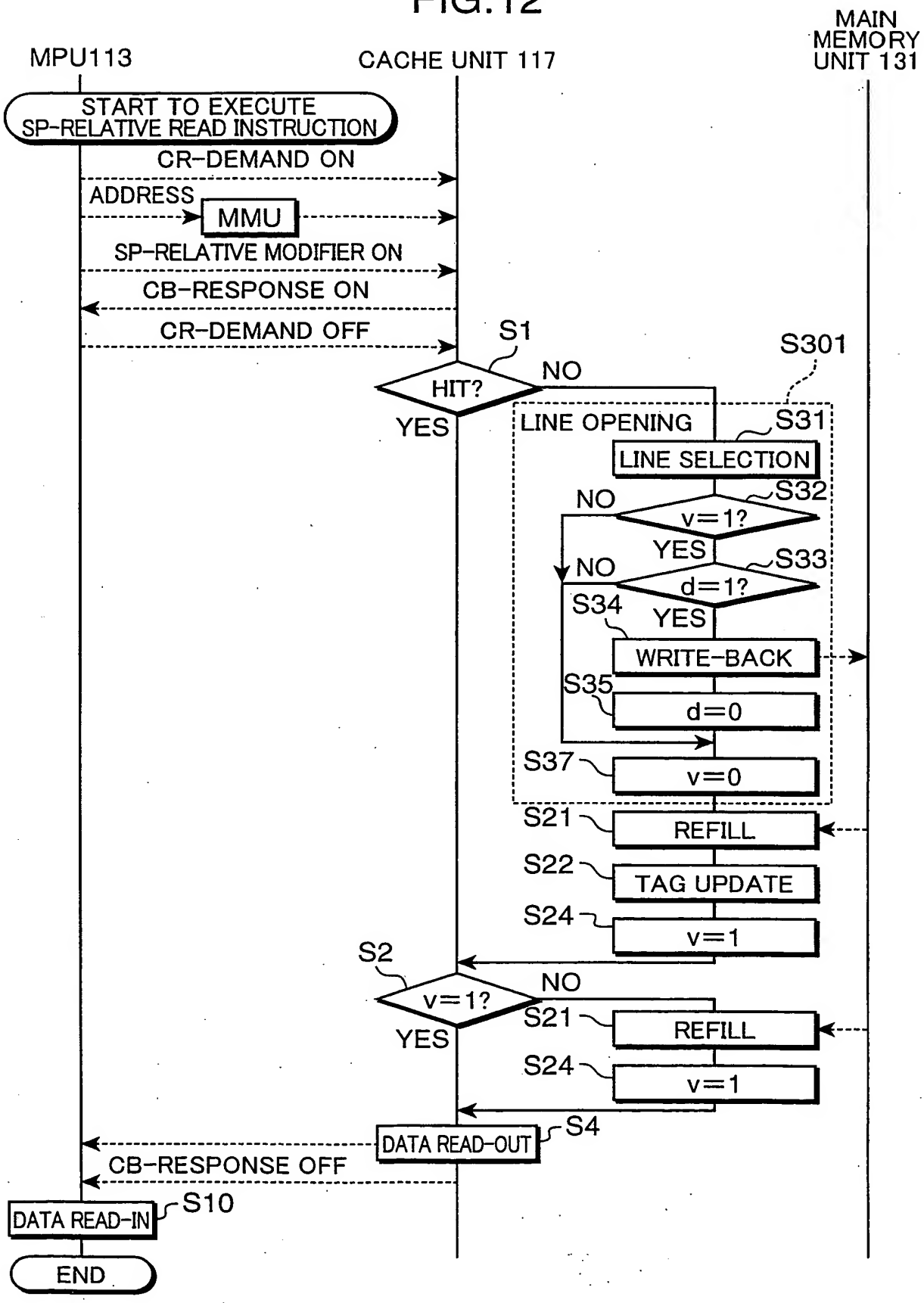

MPU113
CACHE UNIT 117
MAIN MEMORY UNIT 131
START TO EXECUTE SP-RELATIVE READ INSTRUCTION
CR-DEMAND ON
ADDRESS
MMU
SP-RELATIVE MODIFIER ON
CB-RESPONSE ON
CR-DEMAND OFF
S1
HIT?
NO
YES
S301
LINE OPENING
S31
LINE SELECTION
S32
v=1?
NO
YES
S33
d=1?
NO
S34
YES
WRITE-BACK
S35
d=0
S37
v=0
S21
REFILL
S22
TAG UPDATE
S24
v=1
S2
v=1?
NO
YES
S21
REFILL
S24
v=1
S4
DATA READ-OUT
CB-RESPONSE OFF
S10
DATA READ-IN
END

FIG.13

	LINE STATE			OPERATION	
	VALID FLAG	DIRTY FLAG	STACK FLAG	SP-RELATIVE READ	SP-RELATIVE WRITE
(a)	NOT ALLOCATED			DECISION OF THE LINE TO BE OPENED, OPENING OF THE DECIDED LINE, REFILL, OPERATION OF (b)	DECISION OF THE LINE TO BE OPENED, OPENING OF THE DECIDED LINE, REFILL, OPERATION OF (b)
(b)	1	0	0	DATA READ	DATA WRITE d=1
(c)	1	0	1	DATA READ	DATA WRITE d=1
(d)	1	1	0	DATA READ	DATA WRITE
(e)	1	1	1	DATA READ	DATA WRITE
(f)	0	0	0	REFILL v=1 DATA READ	REFILL v=1 DATA WRITE d=1
(g)	0	0	1	REFILL v=1 DATA READ	REFILL v=1 DATA WRITE d=1

FIG.14

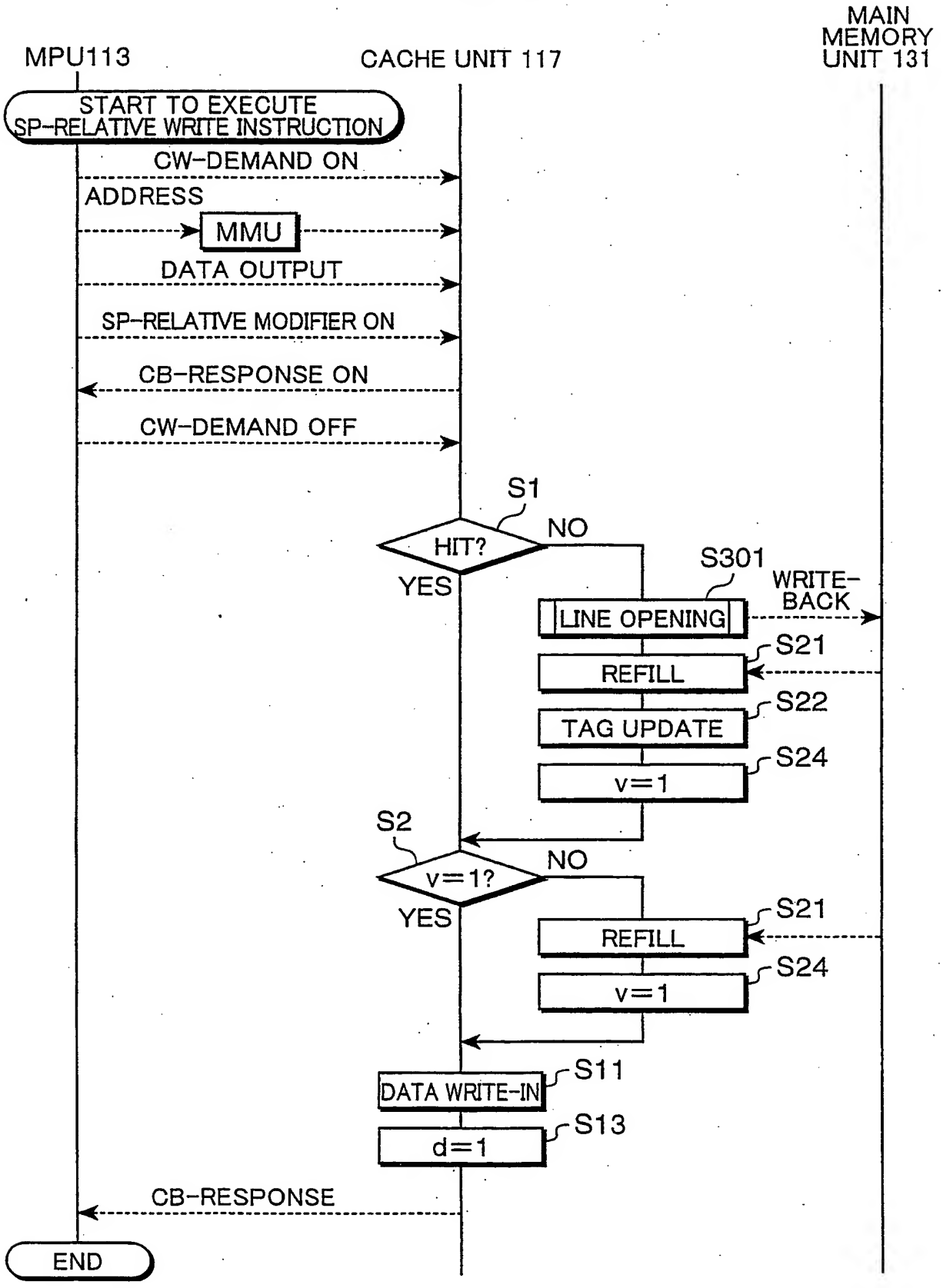

MPU113
CACHE UNIT 117
MAIN MEMORY UNIT 131
START TO EXECUTE SP-RELATIVE WRITE INSTRUCTION
CW-DEMAND ON
ADDRESS
MMU
DATA OUTPUT
SP-RELATIVE MODIFIER ON
CB-RESPONSE ON
CW-DEMAND OFF
S1
HIT?
NO
YES
S301
WRITE-BACK
LINE OPENING
S21
REFILL
S22
TAG UPDATE
S24
v=1
S2
v=1?
NO
YES
S21
REFILL
S24
v=1
S11
DATA WRITE-IN
S13
d=1
CB-RESPONSE
END

FIG.15

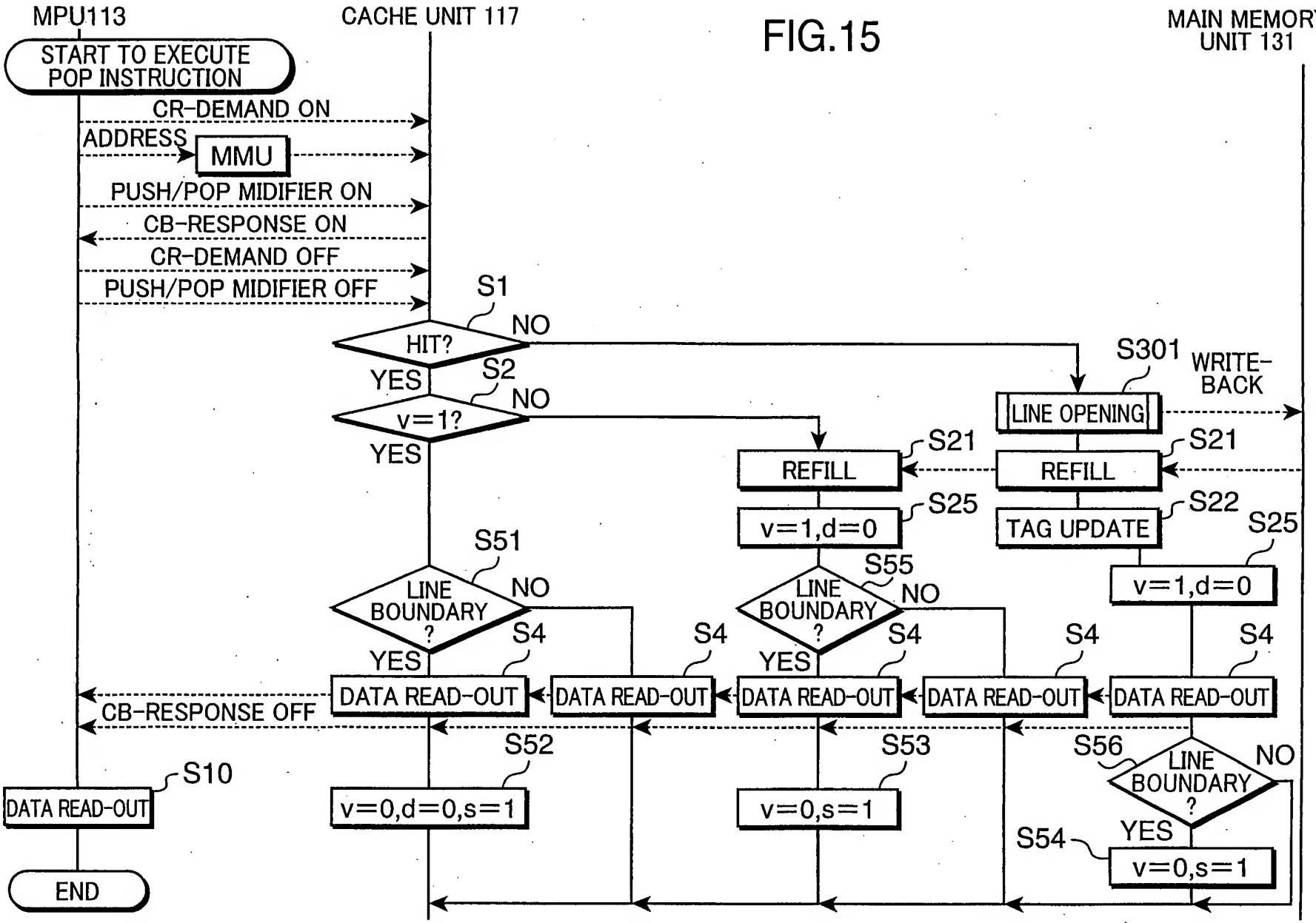

MPU113
CACHE UNIT 117
MAIN MEMORY UNIT 131
START TO EXECUTE POP INSTRUCTION
CR-DEMAND ON
ADDRESS
MMU
PUSH/POP MIDIFIER ON
CB-RESPONSE ON
CR-DEMAND OFF
PUSH/POP MIDIFIER OFF
S1
HIT?
NO
YES
S2
v=1?
NO
YES
S301
WRITE-BACK
LINE OPENING
S21
REFILL
S21
REFILL
S25
v=1,d=0
S22
TAG UPDATE
S25
v=1,d=0
S51
LINE BOUNDARY ?
NO
YES
S55
LINE BOUNDARY ?
NO
YES
S4
DATA READ-OUT
S4
DATA READ-OUT
S4
DATA READ-OUT
S4
DATA READ-OUT
S4
DATA READ-OUT
CB-RESPONSE OFF
S52
v=0,d=0,s=1
S53
v=0,s=1
S56
LINE BOUNDARY ?
NO
YES
S54
v=0,s=1
S10
DATA READ-OUT
END

FIG.16

	LINE STATE			OPERATION	
	VALID FLAG	DIRTY FLAG	STACK FLAG	POP OUT OF THE HIGHEST ADDRESS	POP OUT OF OTHER THAN THE HIGHEST ADDRESS
(a)	NOT ALLOCATED			DECISION OF THE LINE TO BE OPENED, OPENING OF THE DECIDED LINE, REFILL, v=1 d=0 s=1, DATA READ v=0	DECISION OF THE LINE TO BE OPENED, OPENING OF THE DECIDED LINE, REFILL, v=1 d=0, DATA READ
(b)	1	0	0	DATA READ v=0 s=1	DATA READ
(c)	1	0	1	DATA READ v=0	DATA READ
(d)	1	1	0	DATA READ v=0 d=0 s=1	DATA READ
(e)	1	1	1	DATA READ v=0 d=0	DATA READ
(f)	0	0	0	REFILL v=1 DATA READ v=0 s=1	REFILL v=1 DATA READ
(g)	0	0	1	REFILL v=1 DATA READ v=0	REFILL v=1 DATA READ

FIG.17

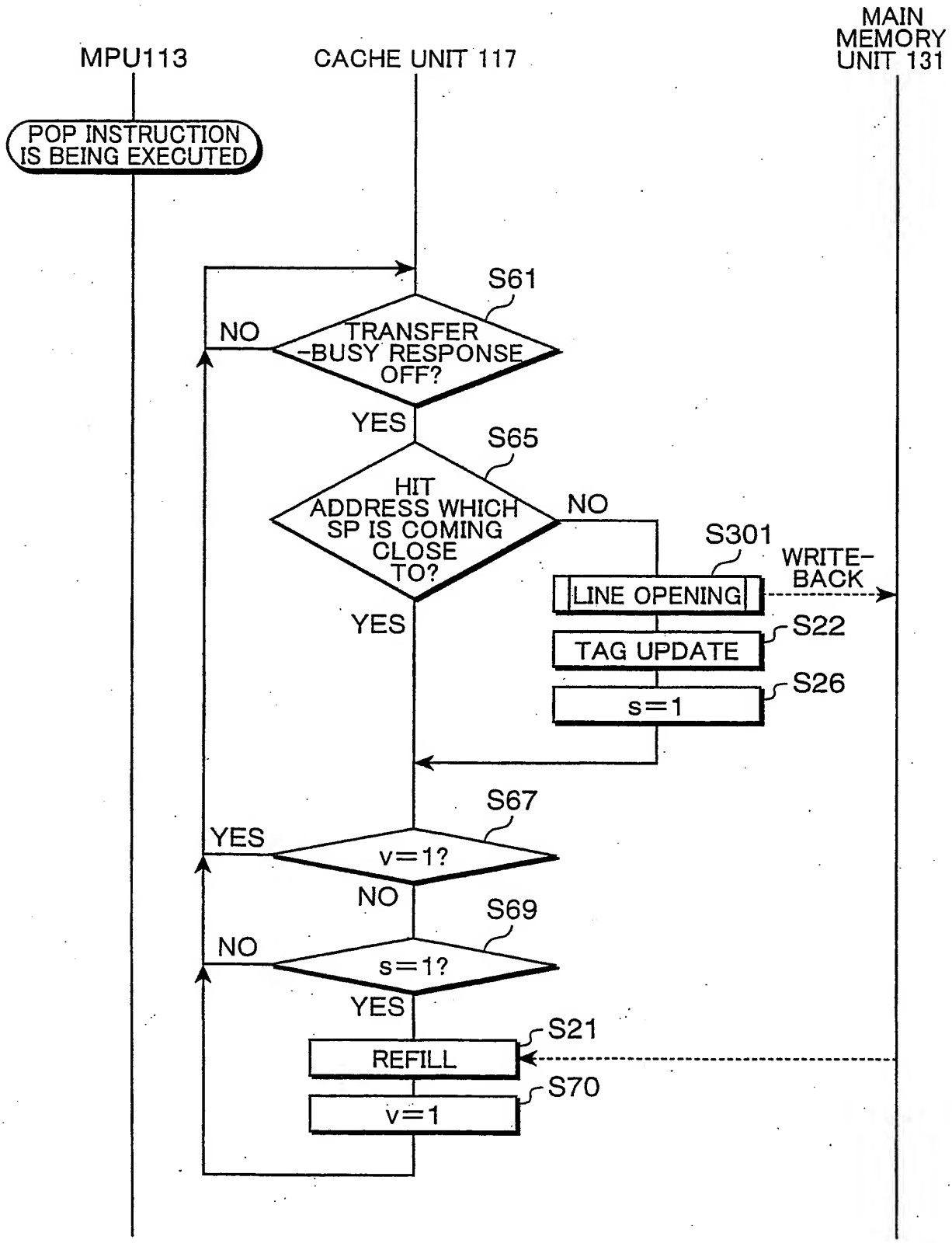

MPU113
CACHE UNIT 117
MAIN MEMORY UNIT 131
POP INSTRUCTION IS BEING EXECUTED
S61
TRANSFER -BUSY RESPONSE OFF?
NO
YES
S65
HIT ADDRESS WHICH SP IS COMING CLOSE TO?
NO
YES
S301
LINE OPENING
WRITE-BACK
S22
TAG UPDATE
S26
s=1
S67
v=1?
YES
NO
S69
s=1?
NO
YES
S21
REFILL
S70
v=1

FIG.18

	LINE STATE			OPERATION	
	VALID FLAG	DIRTY FLAG	STACK FLAG	SP WENT AWAY DOWNWARD (TAG−SP≧12WORDS)	SP CAME CLOSE FROM BELOW (TAG−SP≦8WORDS)
(a)	NOT ALLOCATED				DECISION OF THE LINE TO BE OPENED, OPENING OF THE DECIDED LINE, OPERATION OF (g)
(b)	1	0	0		
(c)	1	0	1		
(d)	1	1	0		
(e)	1	1	1	WRITE-BACK d=0	
(f)	0	0	0		
(g)	0	0	1		REFILL v=1

FIG.19

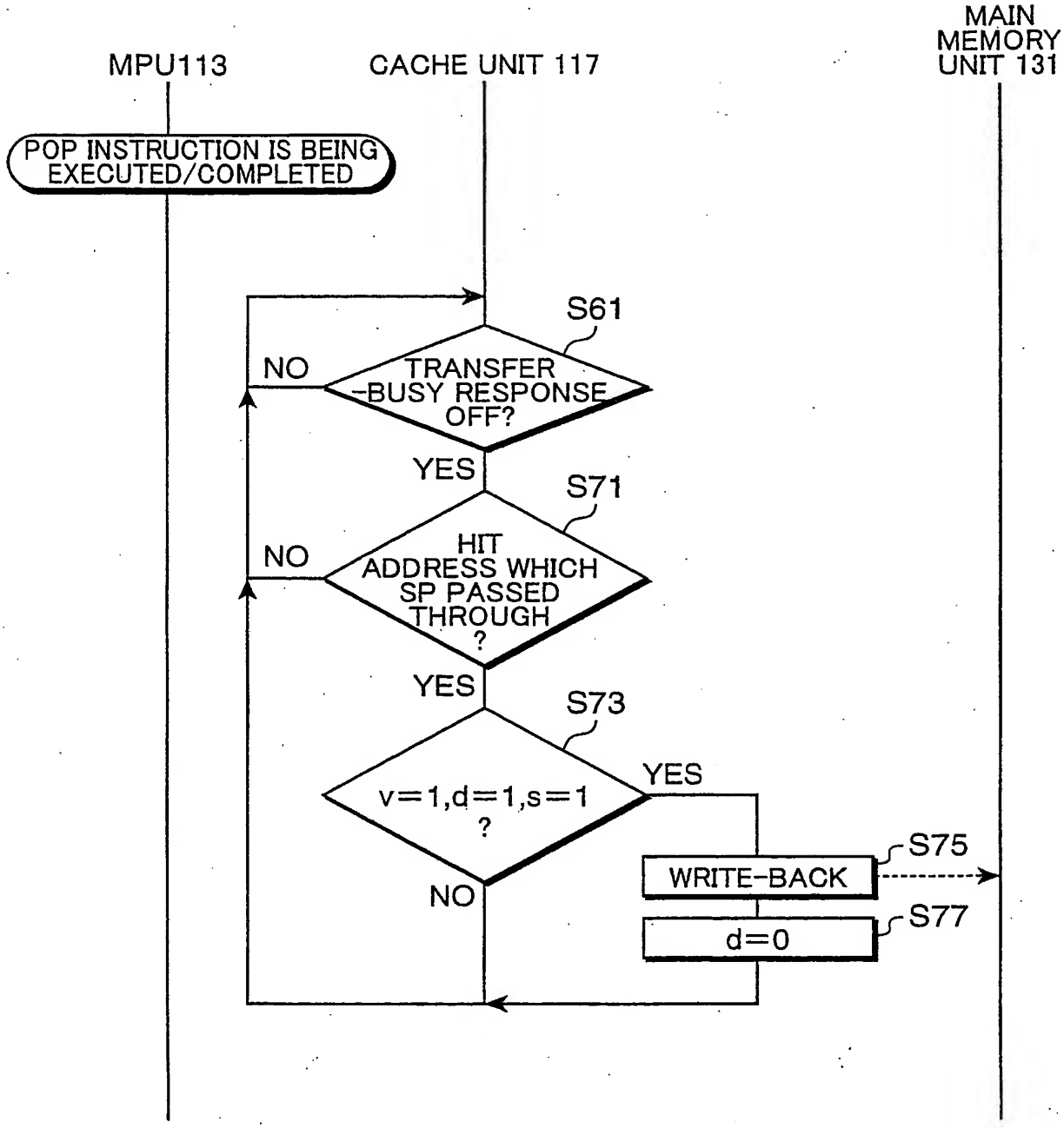

MPU113
CACHE UNIT 117
MAIN MEMORY UNIT 131
POP INSTRUCTION IS BEING EXECUTED/COMPLETED
S61
TRANSFER -BUSY RESPONSE OFF?
NO
YES
S71
HIT ADDRESS WHICH SP PASSED THROUGH ?
NO
YES
S73
v=1,d=1,s=1 ?
YES
NO
S75
WRITE-BACK
S77
d=0

FIG.20

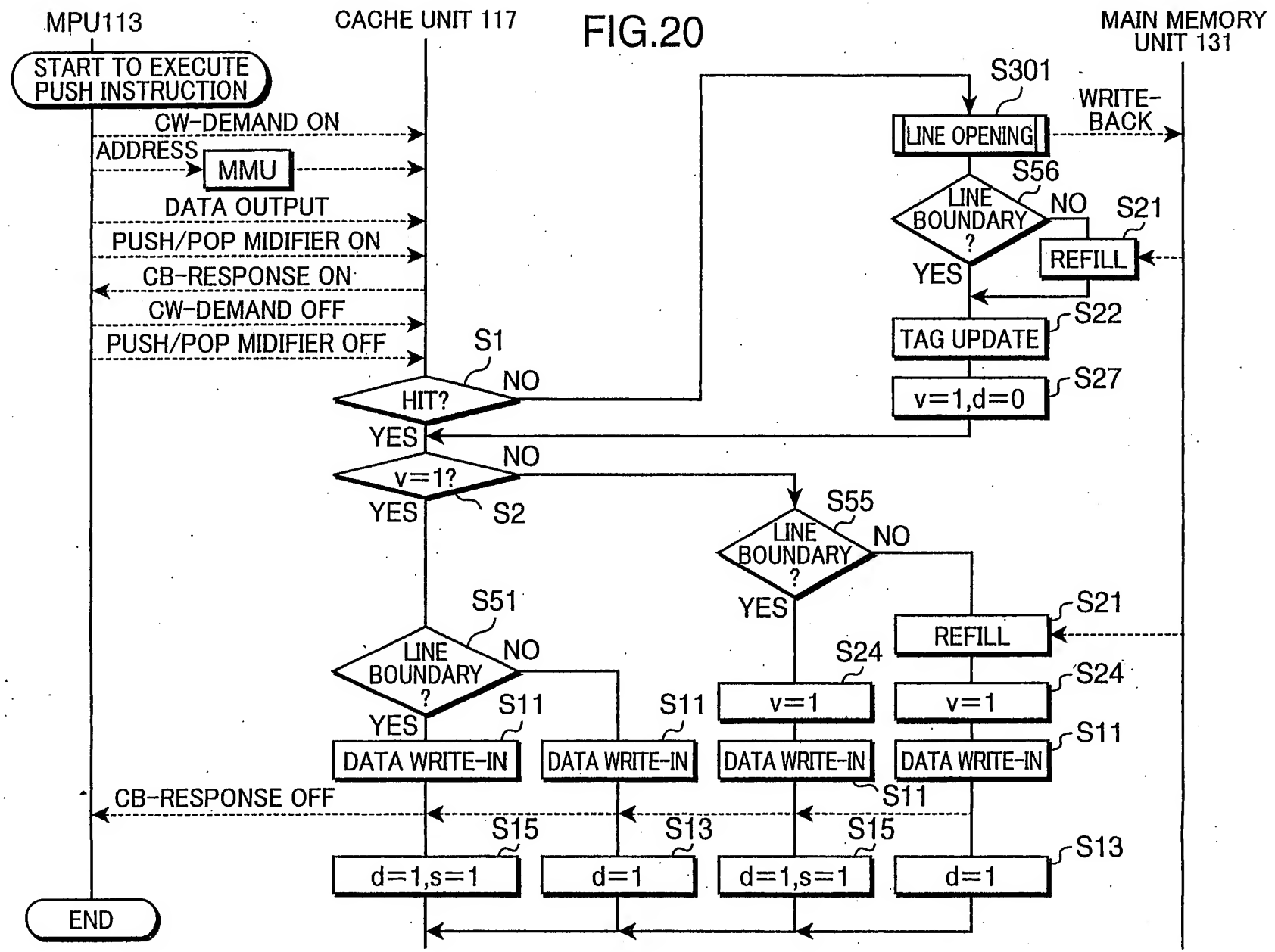

MPU113
CACHE UNIT 117
MAIN MEMORY UNIT 131
START TO EXECUTE PUSH INSTRUCTION
CW-DEMAND ON
ADDRESS
MMU
DATA OUTPUT
PUSH/POP MIDIFIER ON
CB-RESPONSE ON
CW-DEMAND OFF
PUSH/POP MIDIFIER OFF
S1
HIT?
NO
YES
v=1?
NO
YES
S2
S301
LINE OPENING
WRITE-BACK
S56
LINE BOUNDARY ?
NO
YES
S21
REFILL
S22
TAG UPDATE
S27
v=1,d=0
S55
LINE BOUNDARY ?
NO
YES
S51
LINE BOUNDARY ?
NO
YES
S21
REFILL
S24
v=1
S24
v=1
S11
DATA WRITE-IN
S11
DATA WRITE-IN
S11
DATA WRITE-IN
S11
DATA WRITE-IN
S11
CB-RESPONSE OFF
S15
d=1,s=1
S13
d=1
S15
d=1,s=1
S13
d=1
END

FIG.21

	LINE STATE			OPERATION	
	VALID FLAG	DIRTY FLAG	STACK FLAG	PUSH INTO THE HIGHEST ADDRESS	PUSH INTO OTHER THAN THE HIGHEST ADDRESS
(a)	NOT ALLOCATED			DECISION OF THE LINE TO BE OPENED, OPENING OF THE DECIDED LINE, v=1 d=0 s=0, OPERATION OF (b)	DECISION OF THE LINE TO BE OPENED, OPENING OF THE DECIDED LINE, REFILL, v=1 d=0 s=0, OPERATION OF (b)
(b)	1	0	0	DATA WRITE d=1 s=1	DATA WRITE d=1
(c)	1	0	1	DATA WRITE d=1	DATA WRITE d=1
(d)	1	1	0	DATA WRITE s=1	DATA WRITE
(e)	1	1	1	DATA WRITE	DATA WRITE
(f)	0	0	0	v=1 DATA WRITE d=1 s=1	REFILL v=1 DATA WRITE d=1
(g)	0	0	1	v=1 DATA WRITE d=1	REFILL v=1 DATA WRITE d=1

FIG.22

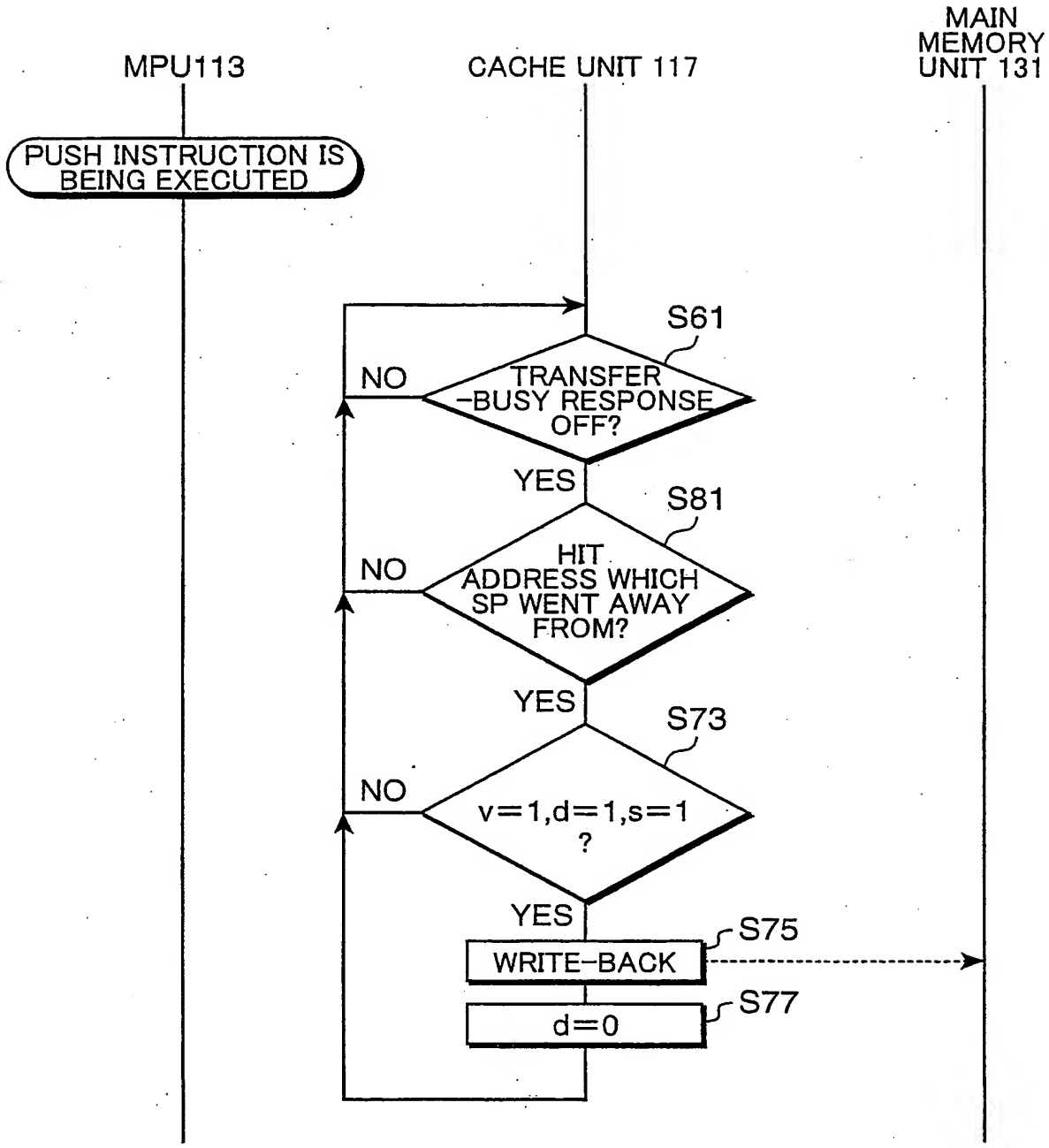

MPU113
CACHE UNIT 117
MAIN MEMORY UNIT 131
PUSH INSTRUCTION IS BEING EXECUTED
S61
TRANSFER -BUSY RESPONSE OFF?
NO
YES
S81
HIT ADDRESS WHICH SP WENT AWAY FROM?
NO
YES
S73
v=1,d=1,s=1 ?
NO
YES
S75
WRITE-BACK
S77
d=0

FIG.23

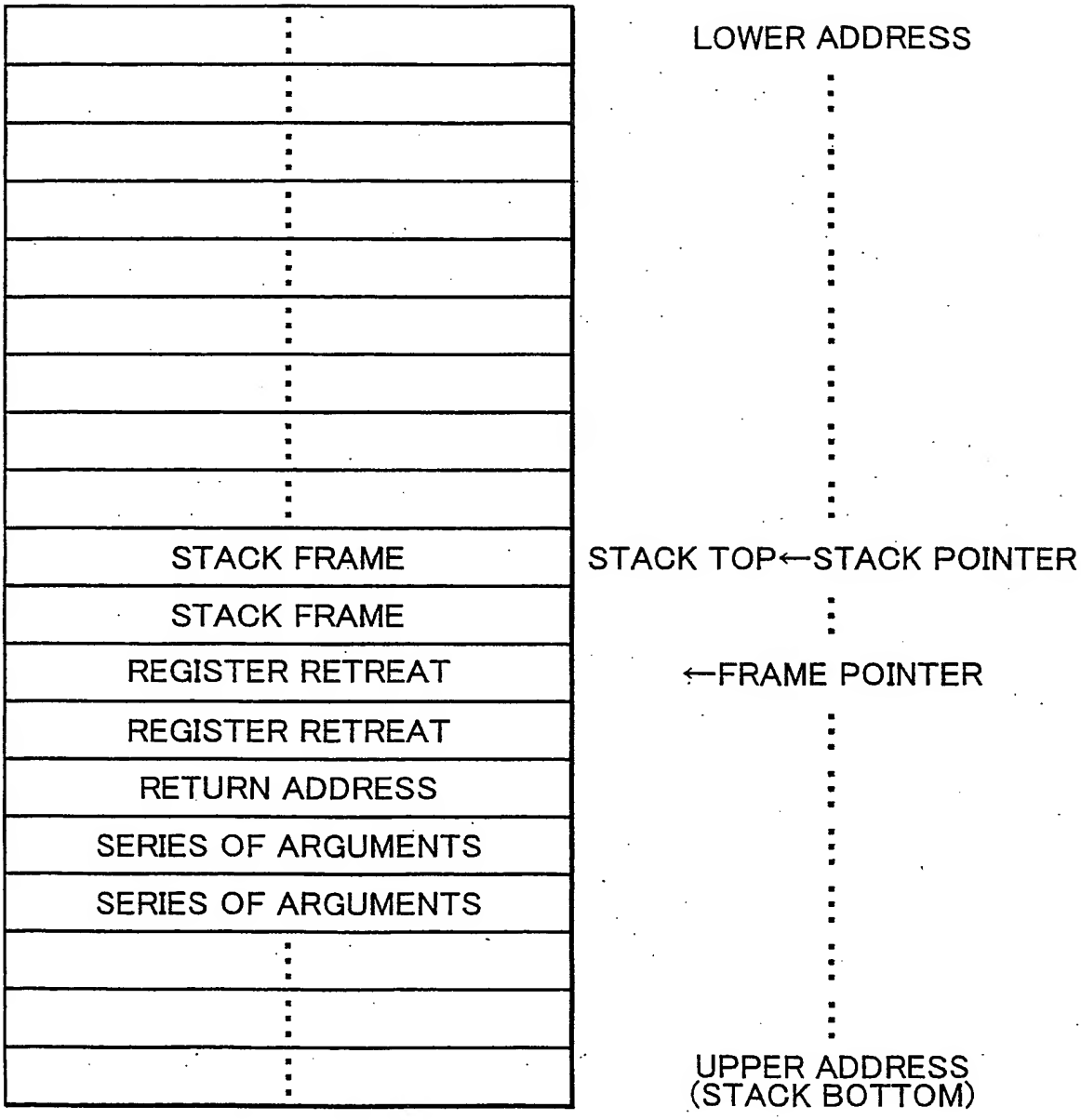

LOWER ADDRESS
STACK FRAME
STACK FRAME
REGISTER RETREAT
REGISTER RETREAT
RETURN ADDRESS
SERIES OF ARGUMENTS
SERIES OF ARGUMENTS
STACK TOP←STACK POINTER
←FRAME POINTER
UPPER ADDRESS
(STACK BOTTOM)

FIG.24

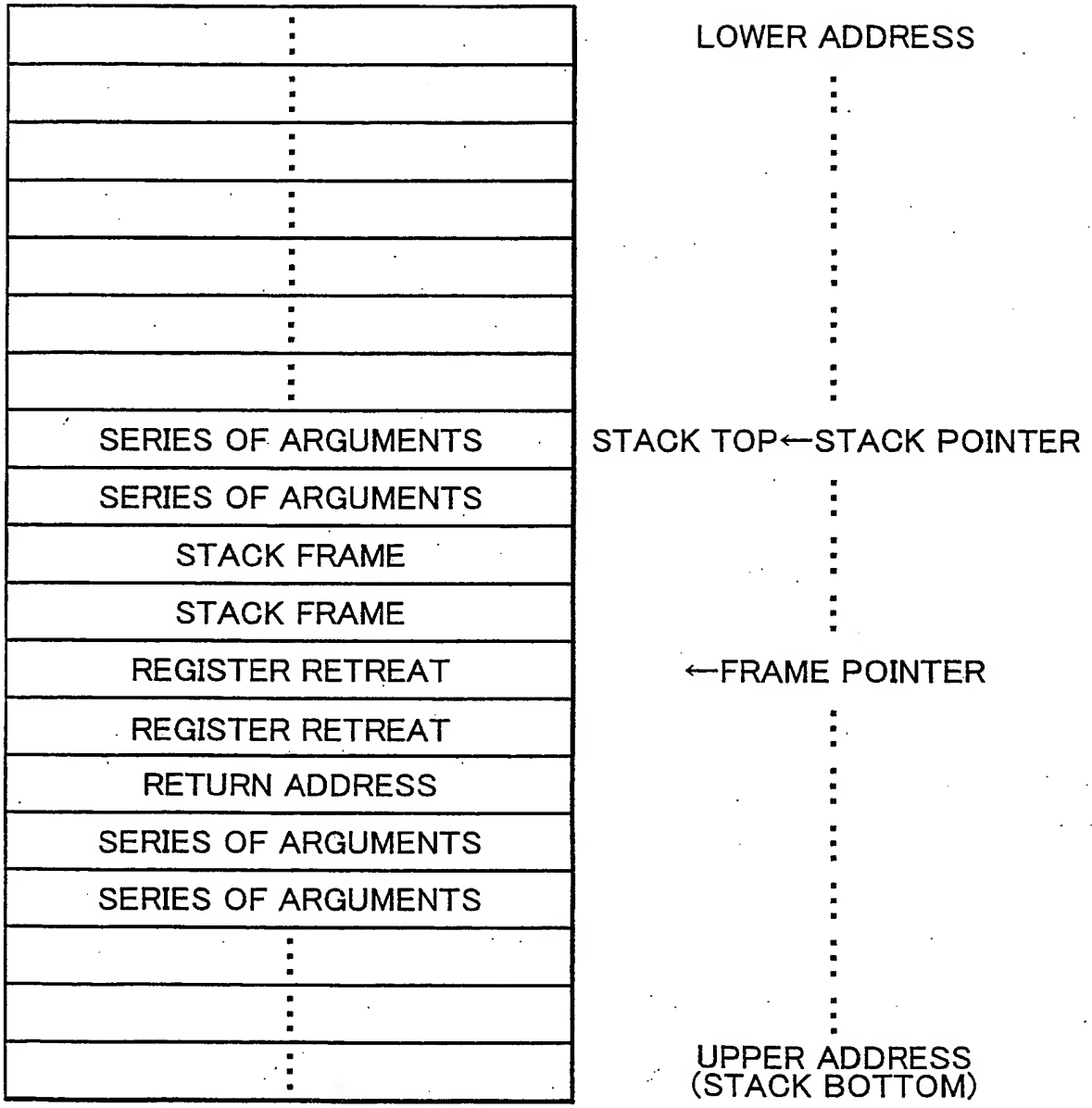

LOWER ADDRESS
SERIES OF ARGUMENTS
STACK TOP←STACK POINTER
SERIES OF ARGUMENTS
STACK FRAME
STACK FRAME
REGISTER RETREAT
←FRAME POINTER
REGISTER RETREAT
RETURN ADDRESS
SERIES OF ARGUMENTS
SERIES OF ARGUMENTS
UPPER ADDRESS
(STACK BOTTOM)

FIG.25

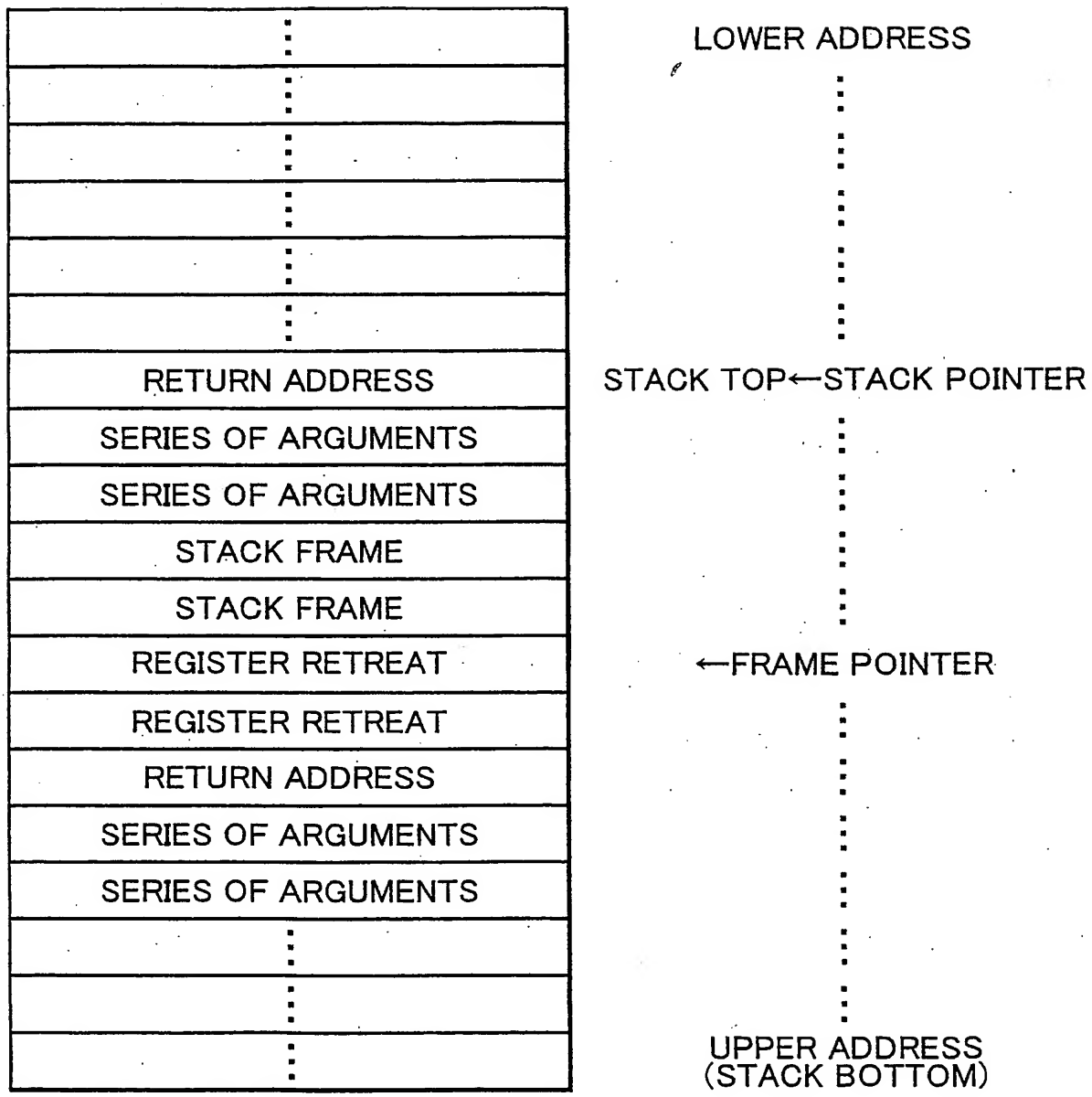

RETURN ADDRESS
SERIES OF ARGUMENTS
SERIES OF ARGUMENTS
STACK FRAME
STACK FRAME
REGISTER RETREAT
REGISTER RETREAT
RETURN ADDRESS
SERIES OF ARGUMENTS
SERIES OF ARGUMENTS
LOWER ADDRESS
STACK TOP←STACK POINTER
←FRAME POINTER
UPPER ADDRESS
(STACK BOTTOM)

FIG.26

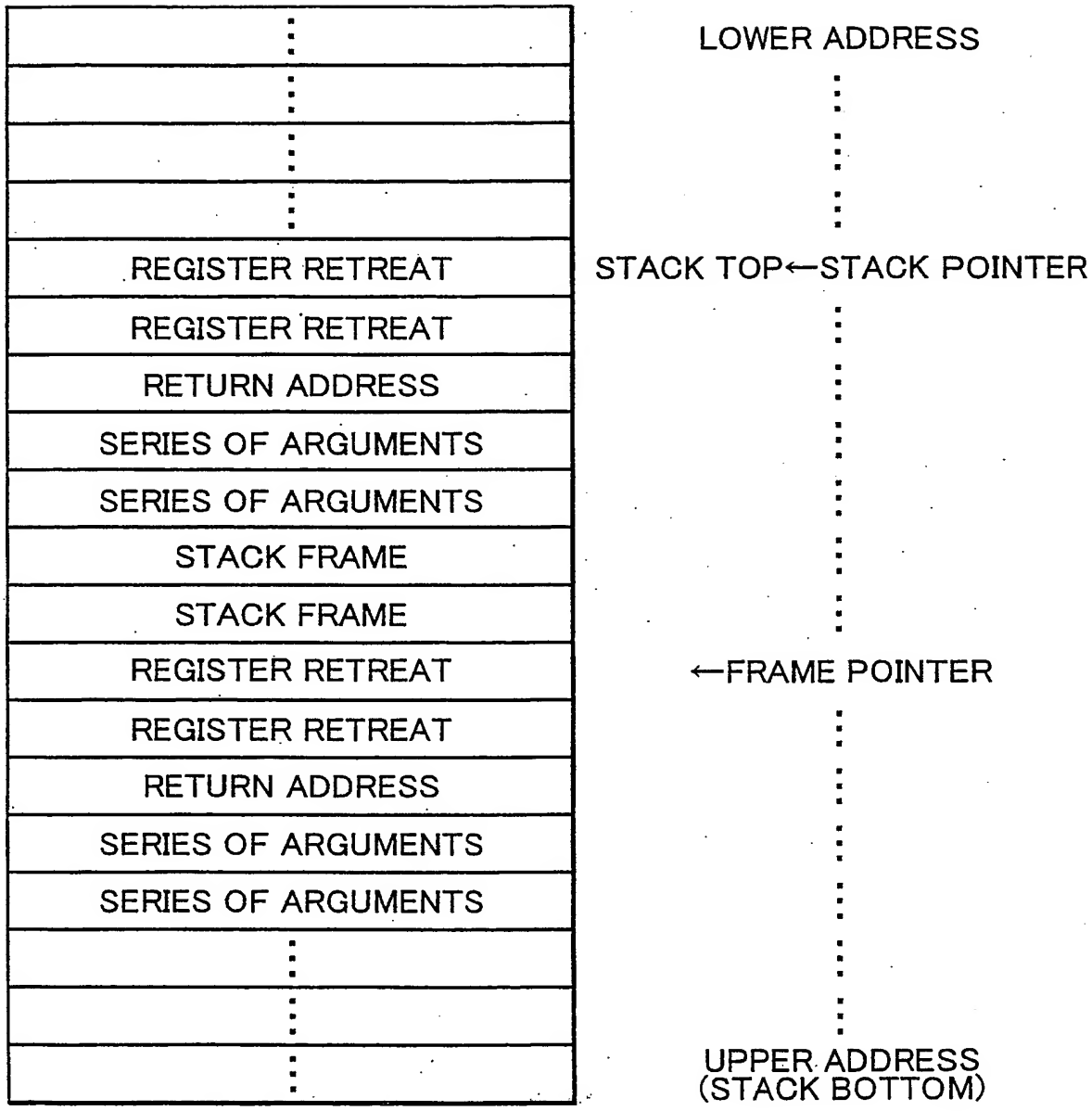

REGISTER RETREAT
REGISTER RETREAT
RETURN ADDRESS
SERIES OF ARGUMENTS
SERIES OF ARGUMENTS
STACK FRAME
STACK FRAME
REGISTER RETREAT
REGISTER RETREAT
RETURN ADDRESS
SERIES OF ARGUMENTS
SERIES OF ARGUMENTS
LOWER ADDRESS
STACK TOP←STACK POINTER
←FRAME POINTER
UPPER ADDRESS
(STACK BOTTOM)

FIG.27

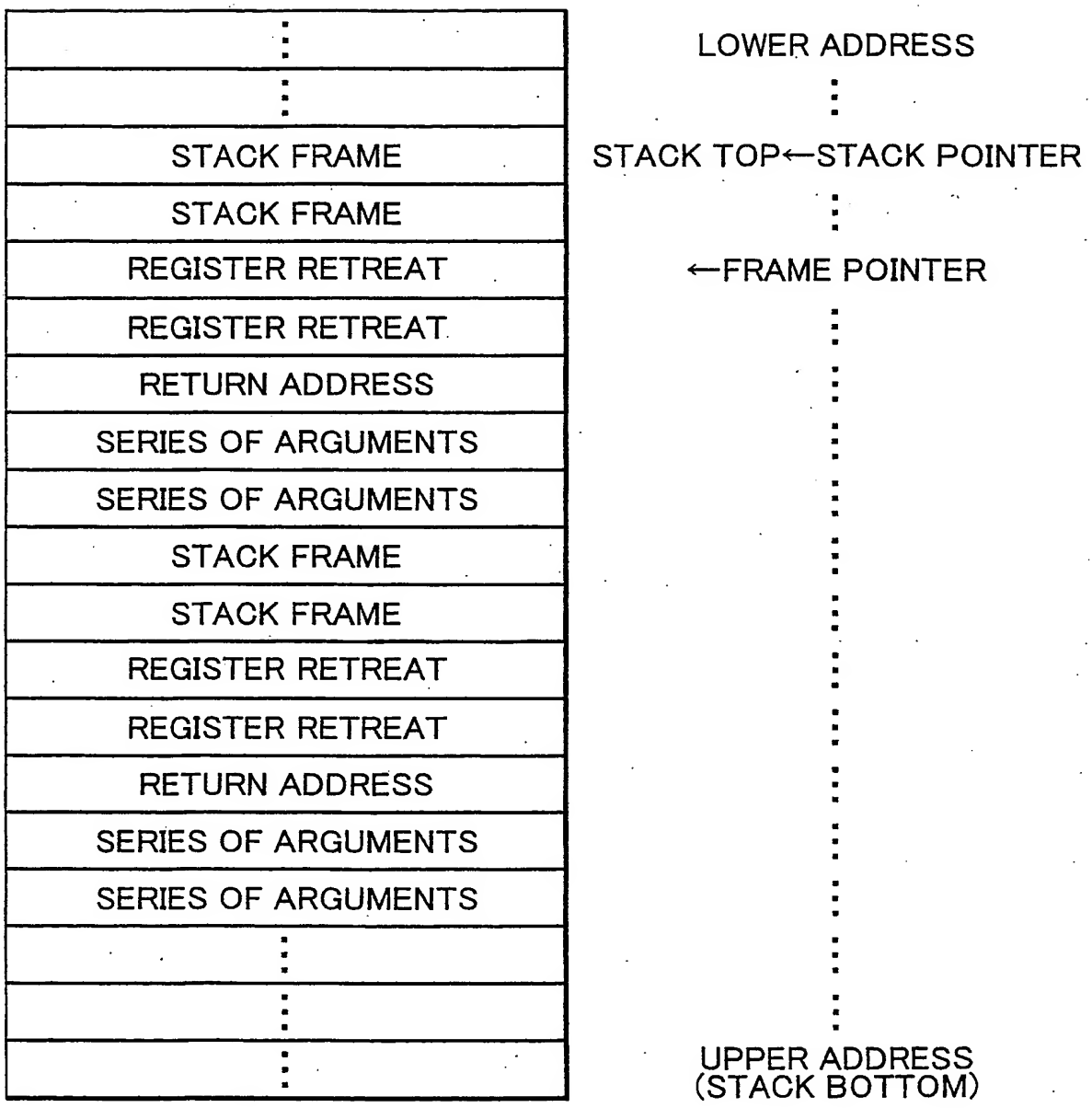

STACK FRAME
STACK FRAME
REGISTER RETREAT
REGISTER RETREAT
RETURN ADDRESS
SERIES OF ARGUMENTS
SERIES OF ARGUMENTS
STACK FRAME
STACK FRAME
REGISTER RETREAT
REGISTER RETREAT
RETURN ADDRESS
SERIES OF ARGUMENTS
SERIES OF ARGUMENTS
LOWER ADDRESS
STACK TOP←STACK POINTER
←FRAME POINTER
UPPER ADDRESS
(STACK BOTTOM)

FIG.28

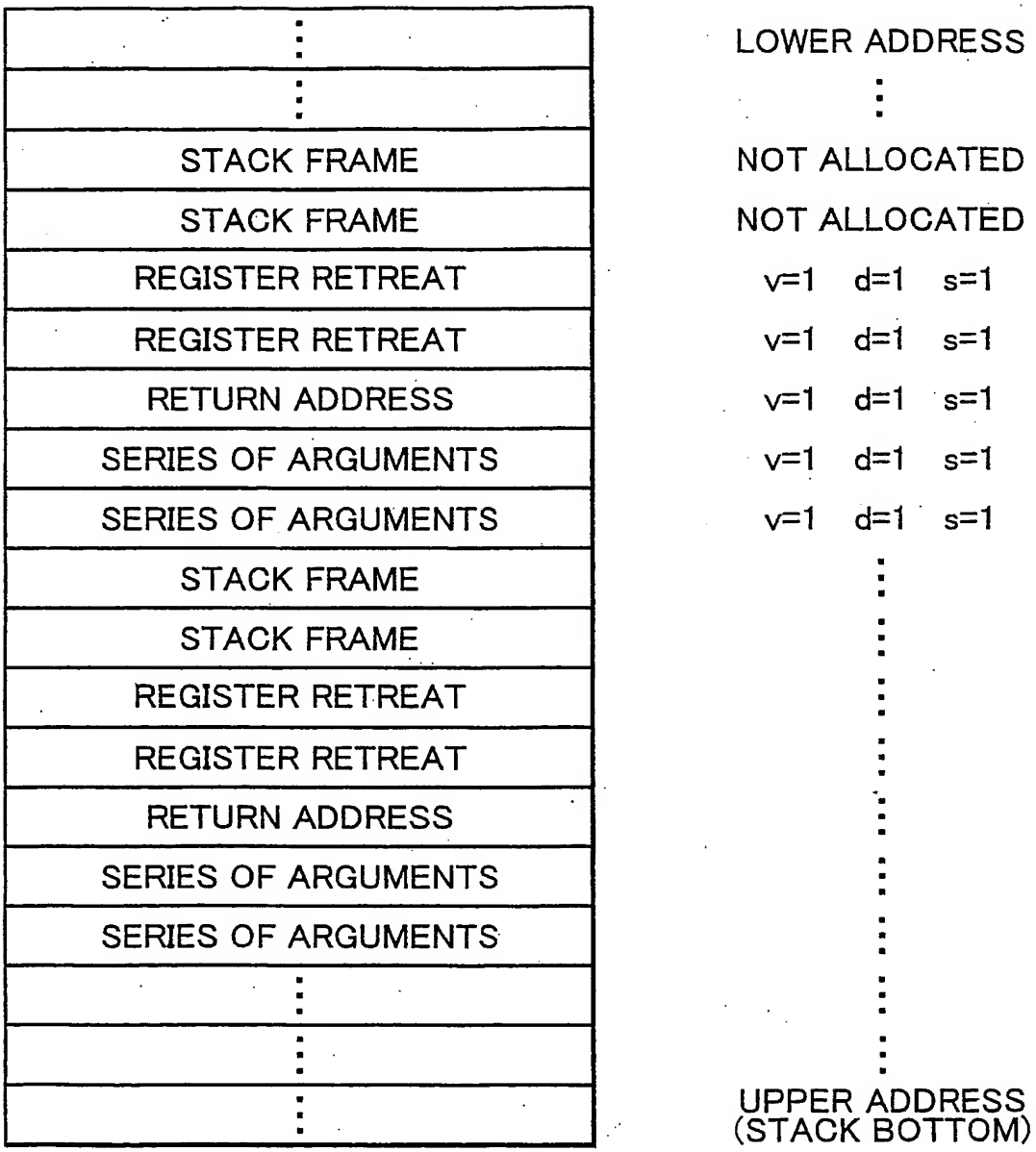

STACK FRAME
STACK FRAME
REGISTER RETREAT
REGISTER RETREAT
RETURN ADDRESS
SERIES OF ARGUMENTS
SERIES OF ARGUMENTS
STACK FRAME
STACK FRAME
REGISTER RETREAT
REGISTER RETREAT
RETURN ADDRESS
SERIES OF ARGUMENTS
SERIES OF ARGUMENTS
LOWER ADDRESS
NOT ALLOCATED
NOT ALLOCATED
v=1 d=1 s=1
v=1 d=1 s=1
v=1 d=1 s=1
v=1 d=1 s=1
v=1 d=1 s=1
UPPER ADDRESS
(STACK BOTTOM)

FIG.29

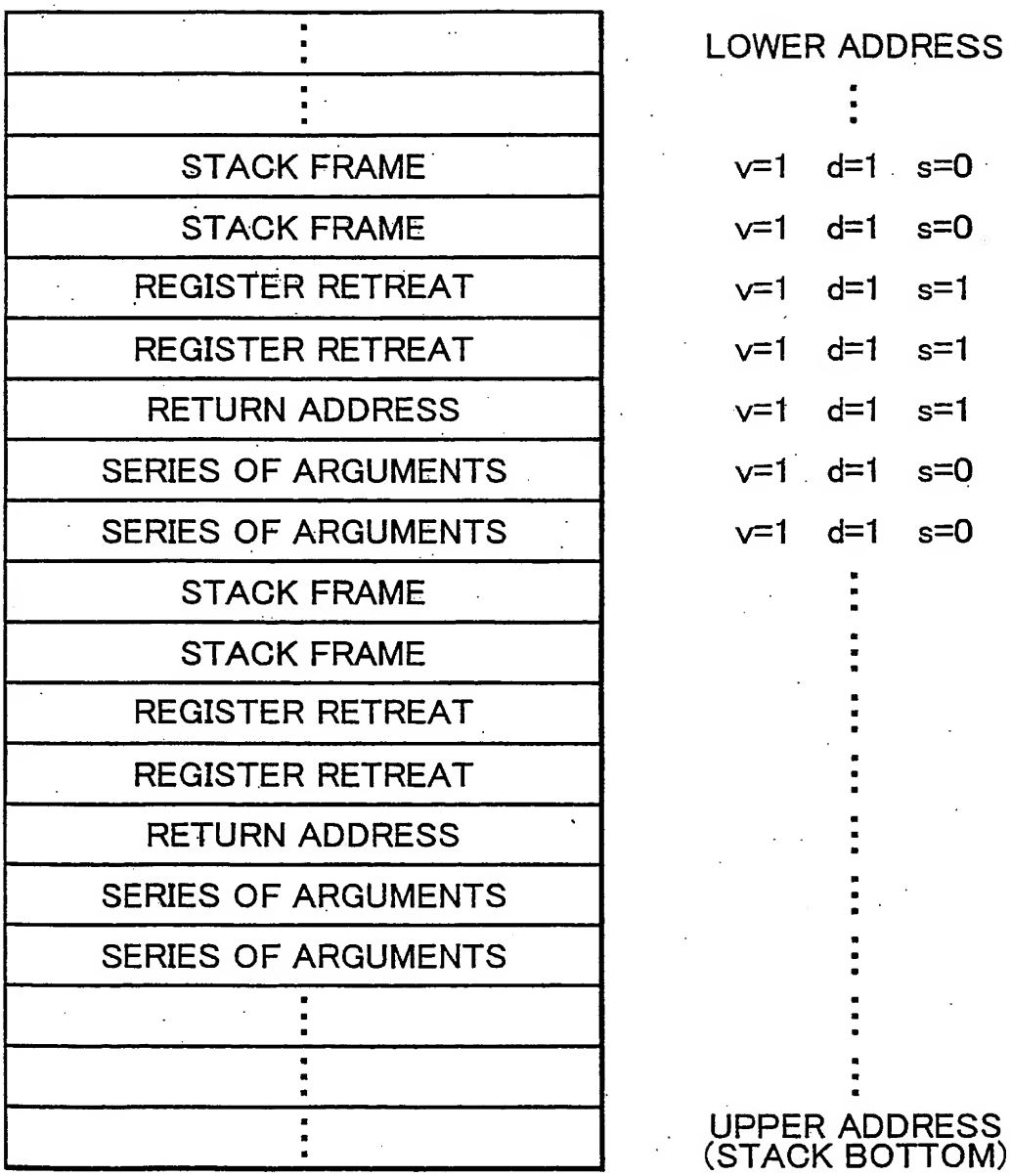

STACK FRAME
STACK FRAME
REGISTER RETREAT
REGISTER RETREAT
RETURN ADDRESS
SERIES OF ARGUMENTS
SERIES OF ARGUMENTS
STACK FRAME
STACK FRAME
REGISTER RETREAT
REGISTER RETREAT
RETURN ADDRESS
SERIES OF ARGUMENTS
SERIES OF ARGUMENTS
LOWER ADDRESS
v=1 d=1 s=0
v=1 d=1 s=0
v=1 d=1 s=1
v=1 d=1 s=1
v=1 d=1 s=1
v=1 d=1 s=0
v=1 d=1 s=0
UPPER ADDRESS
(STACK BOTTOM)

FIG.30

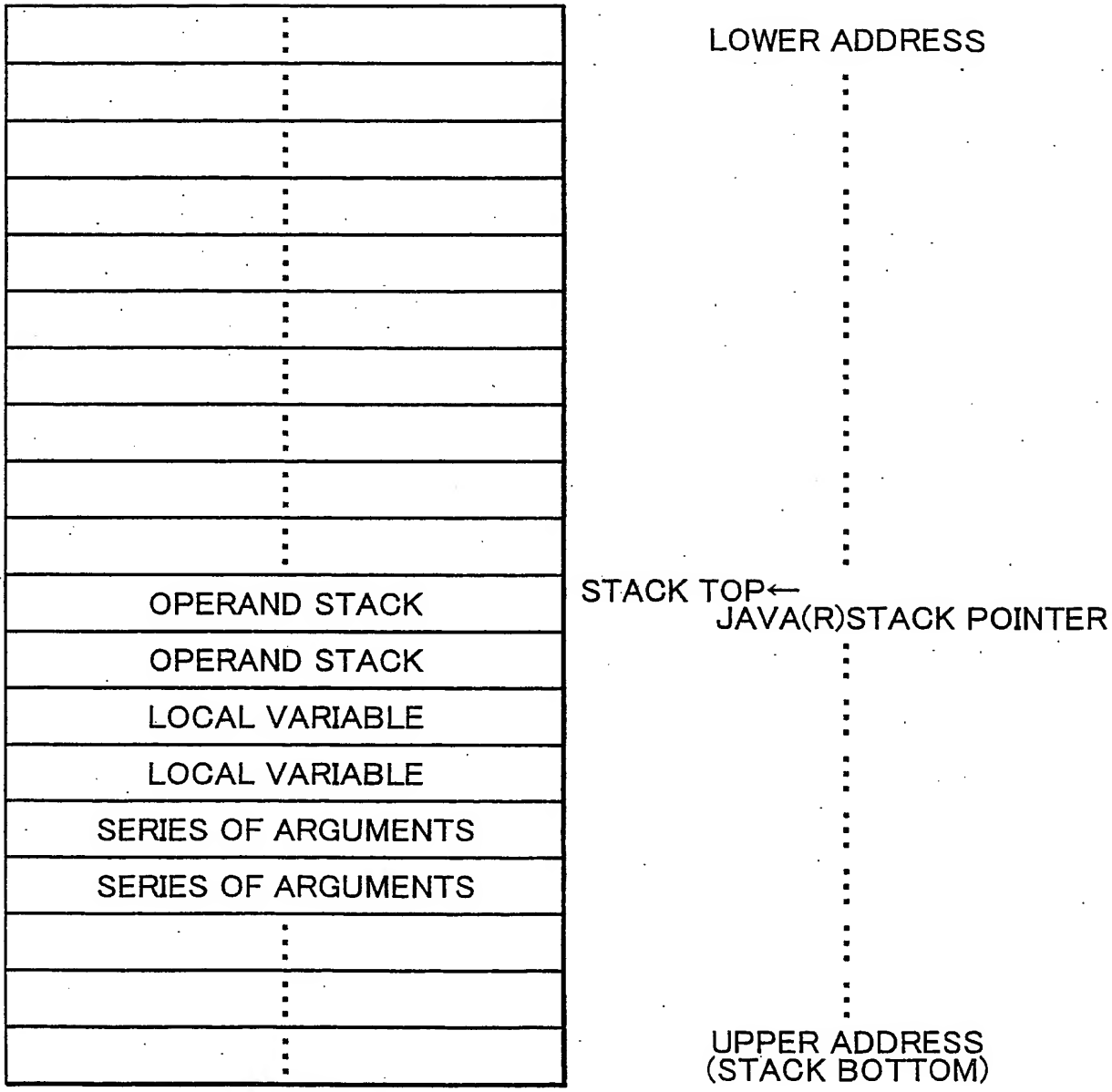

LOWER ADDRESS
OPERAND STACK
OPERAND STACK
LOCAL VARIABLE
LOCAL VARIABLE
SERIES OF ARGUMENTS
SERIES OF ARGUMENTS
STACK TOP←
JAVA(R)STACK POINTER
UPPER ADDRESS
(STACK BOTTOM)

FIG.31

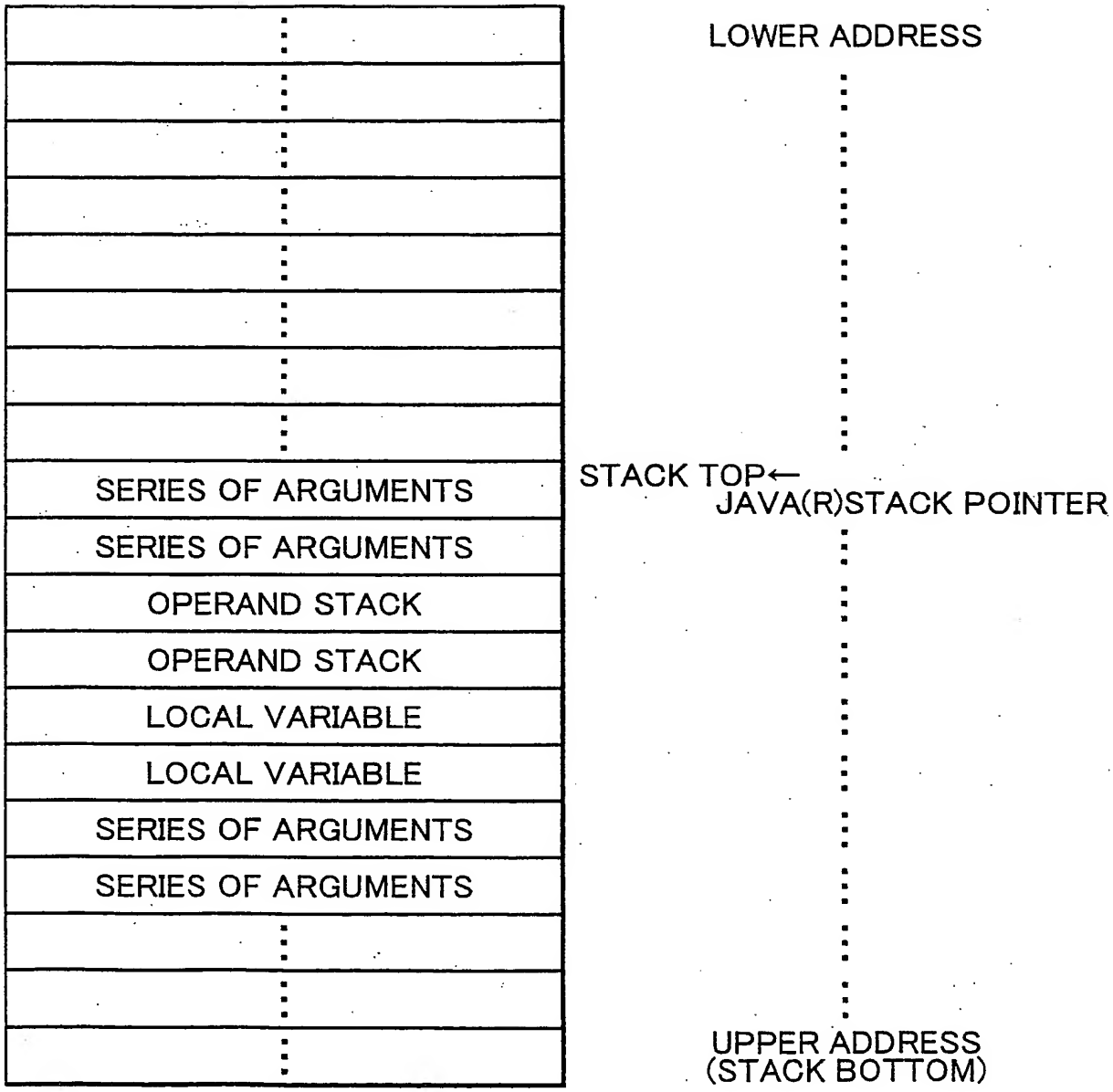

LOWER ADDRESS
STACK TOP←
JAVA(R)STACK POINTER
SERIES OF ARGUMENTS
SERIES OF ARGUMENTS
OPERAND STACK
OPERAND STACK
LOCAL VARIABLE
LOCAL VARIABLE
SERIES OF ARGUMENTS
SERIES OF ARGUMENTS
UPPER ADDRESS
(STACK BOTTOM)

FIG.32

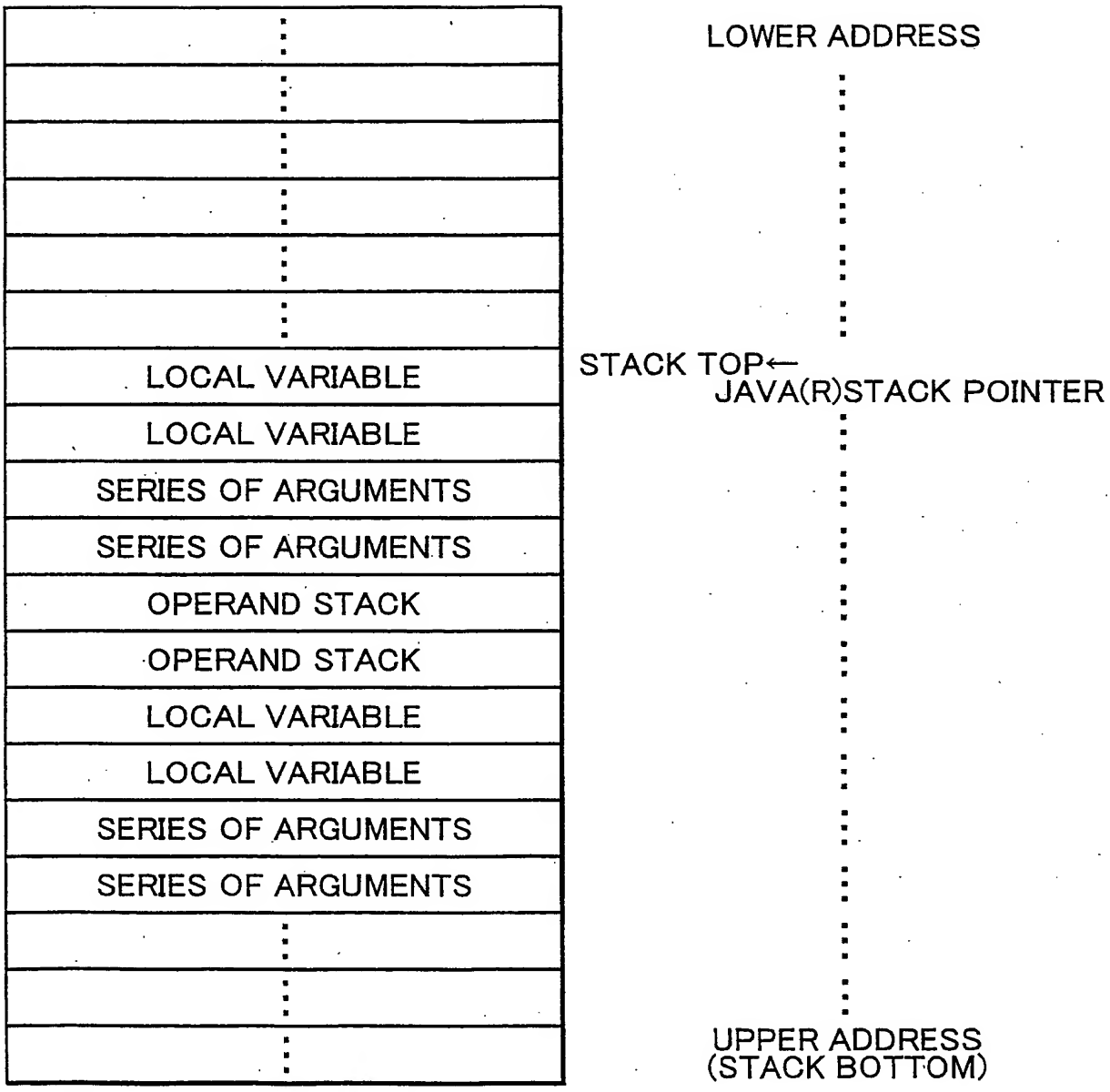

LOWER ADDRESS
STACK TOP←
JAVA(R)STACK POINTER
LOCAL VARIABLE
LOCAL VARIABLE
SERIES OF ARGUMENTS
SERIES OF ARGUMENTS
OPERAND STACK
OPERAND STACK
LOCAL VARIABLE
LOCAL VARIABLE
SERIES OF ARGUMENTS
SERIES OF ARGUMENTS
UPPER ADDRESS
(STACK BOTTOM)

FIG.33

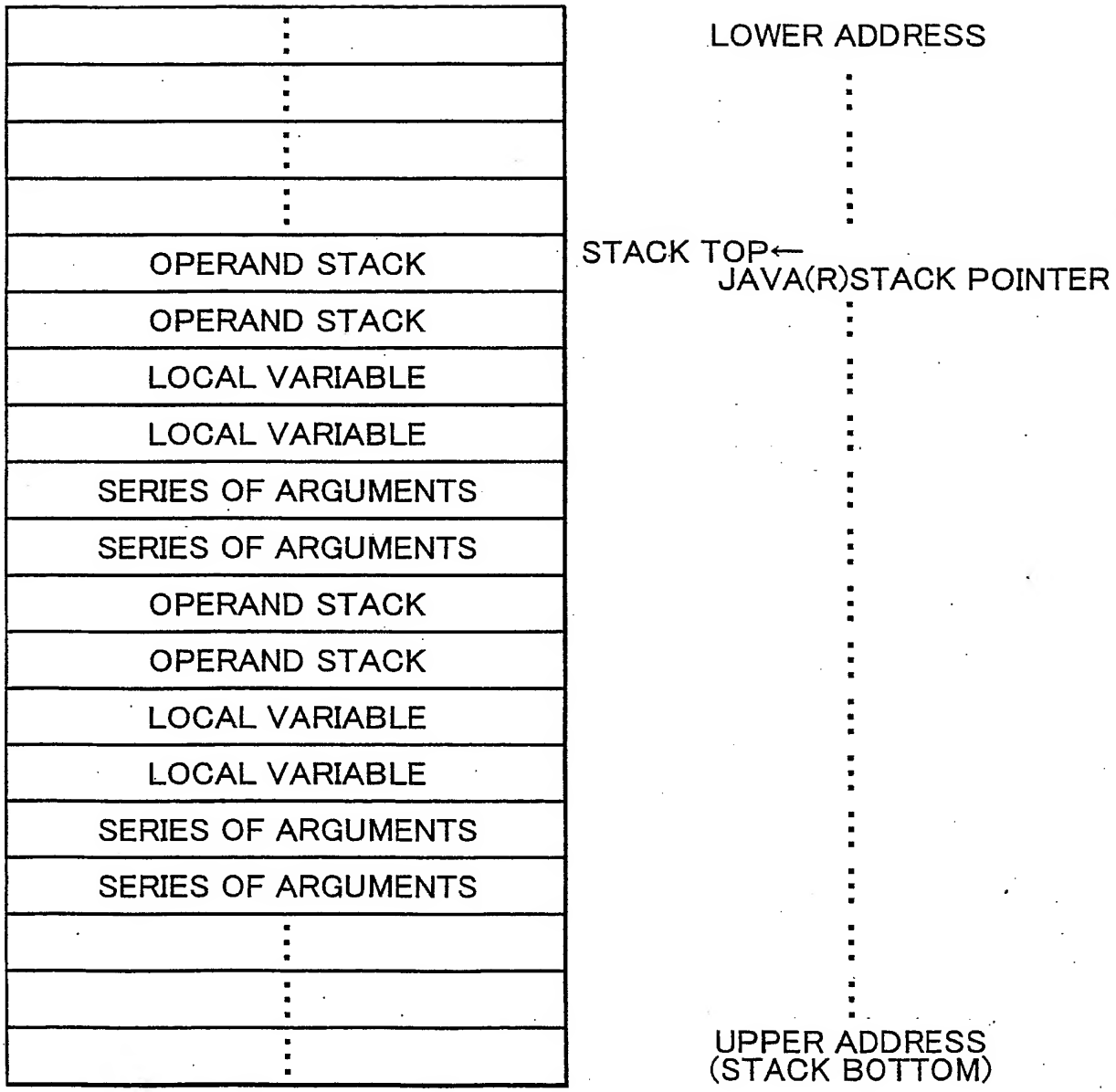

OPERAND STACK
OPERAND STACK
LOCAL VARIABLE
LOCAL VARIABLE
SERIES OF ARGUMENTS
SERIES OF ARGUMENTS
OPERAND STACK
OPERAND STACK
LOCAL VARIABLE
LOCAL VARIABLE
SERIES OF ARGUMENTS
SERIES OF ARGUMENTS
LOWER ADDRESS
STACK TOP←
JAVA(R)STACK POINTER
UPPER ADDRESS
(STACK BOTTOM)

FIG.34

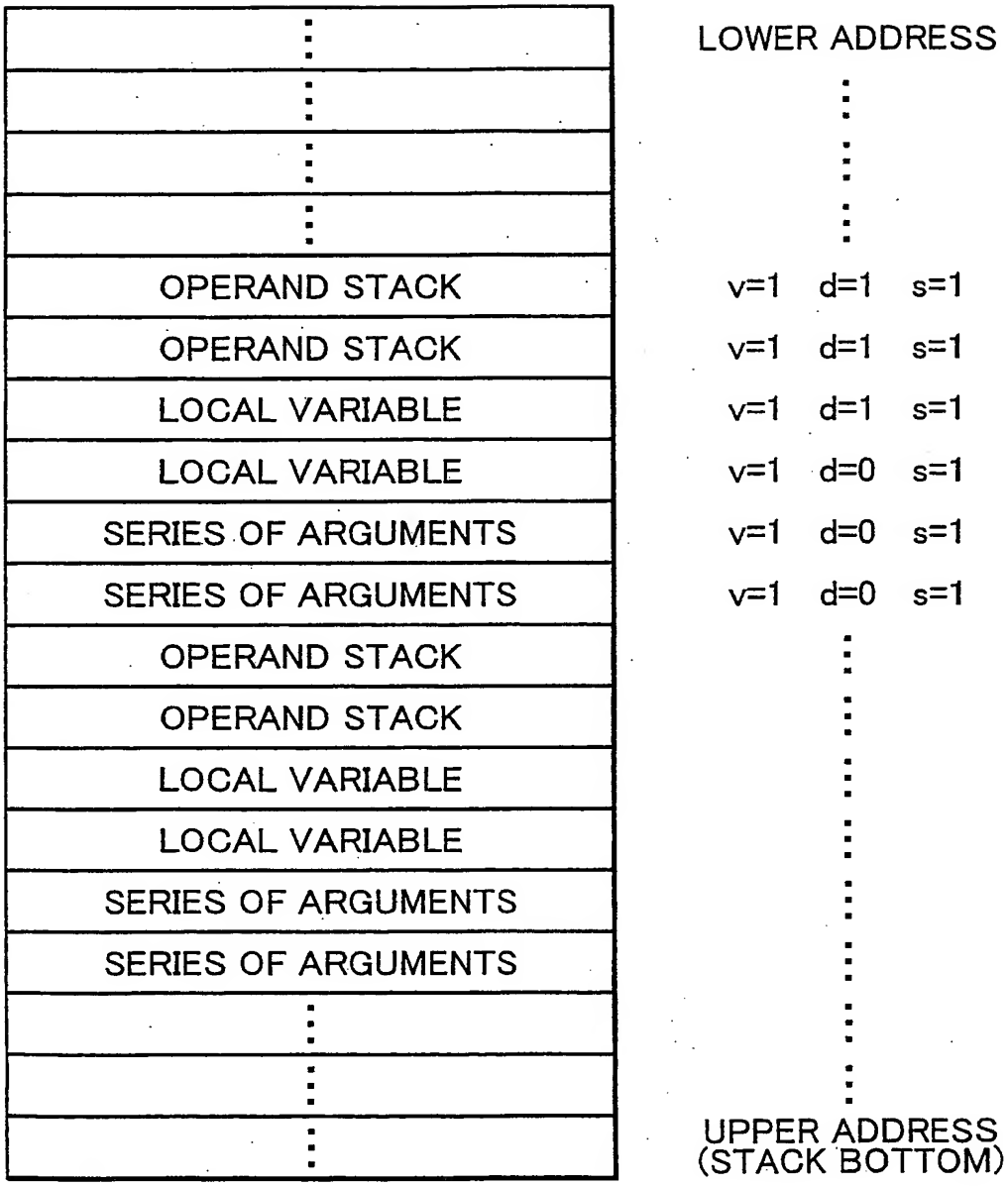

LOWER ADDRESS
OPERAND STACK
OPERAND STACK
LOCAL VARIABLE
LOCAL VARIABLE
SERIES OF ARGUMENTS
SERIES OF ARGUMENTS
OPERAND STACK
OPERAND STACK
LOCAL VARIABLE
LOCAL VARIABLE
SERIES OF ARGUMENTS
SERIES OF ARGUMENTS
v=1 d=1 s=1
v=1 d=1 s=1
v=1 d=1 s=1
v=1 d=0 s=1
v=1 d=0 s=1
v=1 d=0 s=1
UPPER ADDRESS
(STACK BOTTOM)

FIG.35

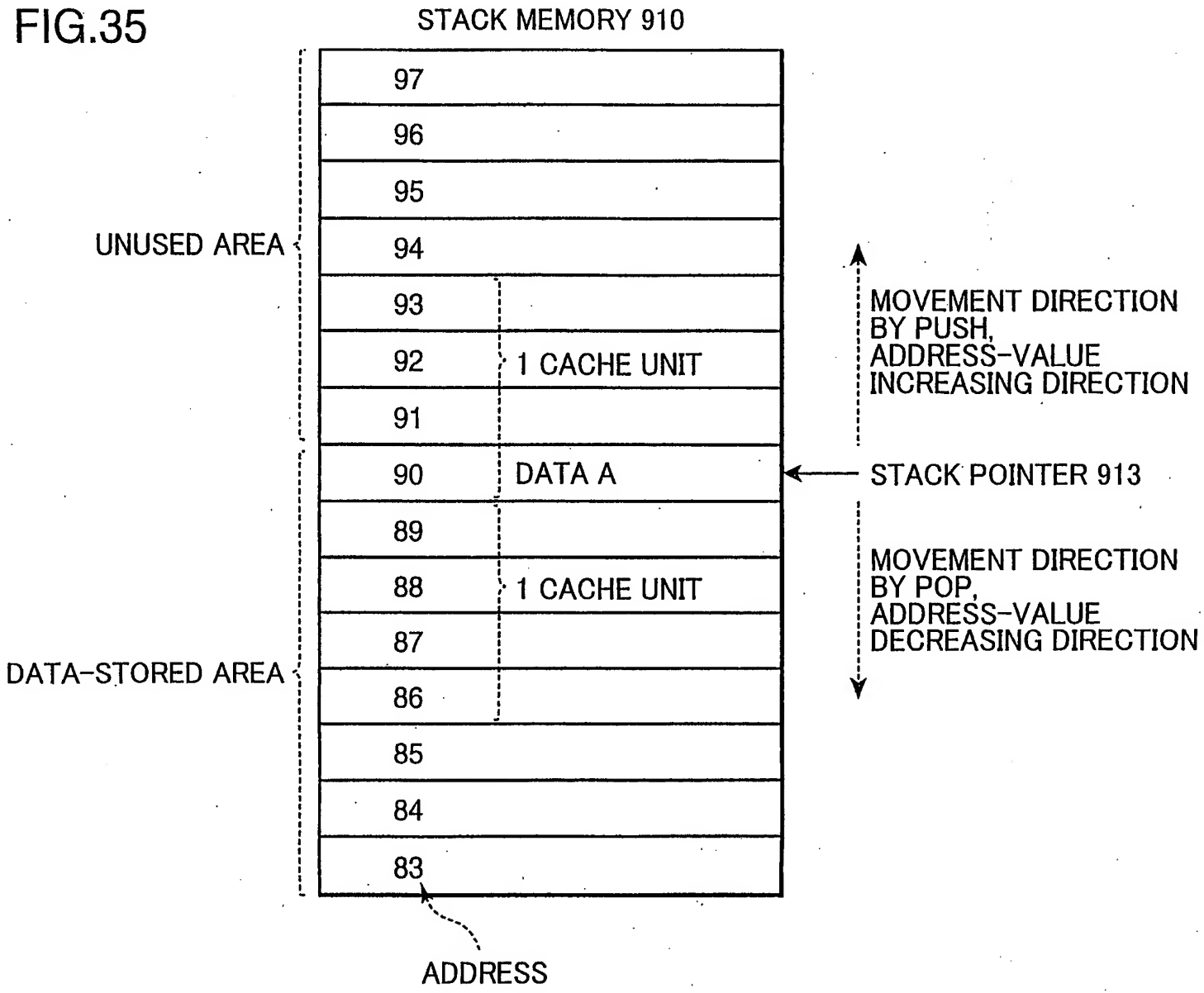

STACK MEMORY 910
97
96
95
94
93
92
91
90
89
88
87
86
85
84
83
UNUSED AREA
DATA-STORED AREA
1 CACHE UNIT
1 CACHE UNIT
DATA A
STACK POINTER 913
MOVEMENT DIRECTION BY PUSH, ADDRESS-VALUE INCREASING DIRECTION
MOVEMENT DIRECTION BY POP, ADDRESS-VALUE DECREASING DIRECTION
ADDRESS

FIG.36

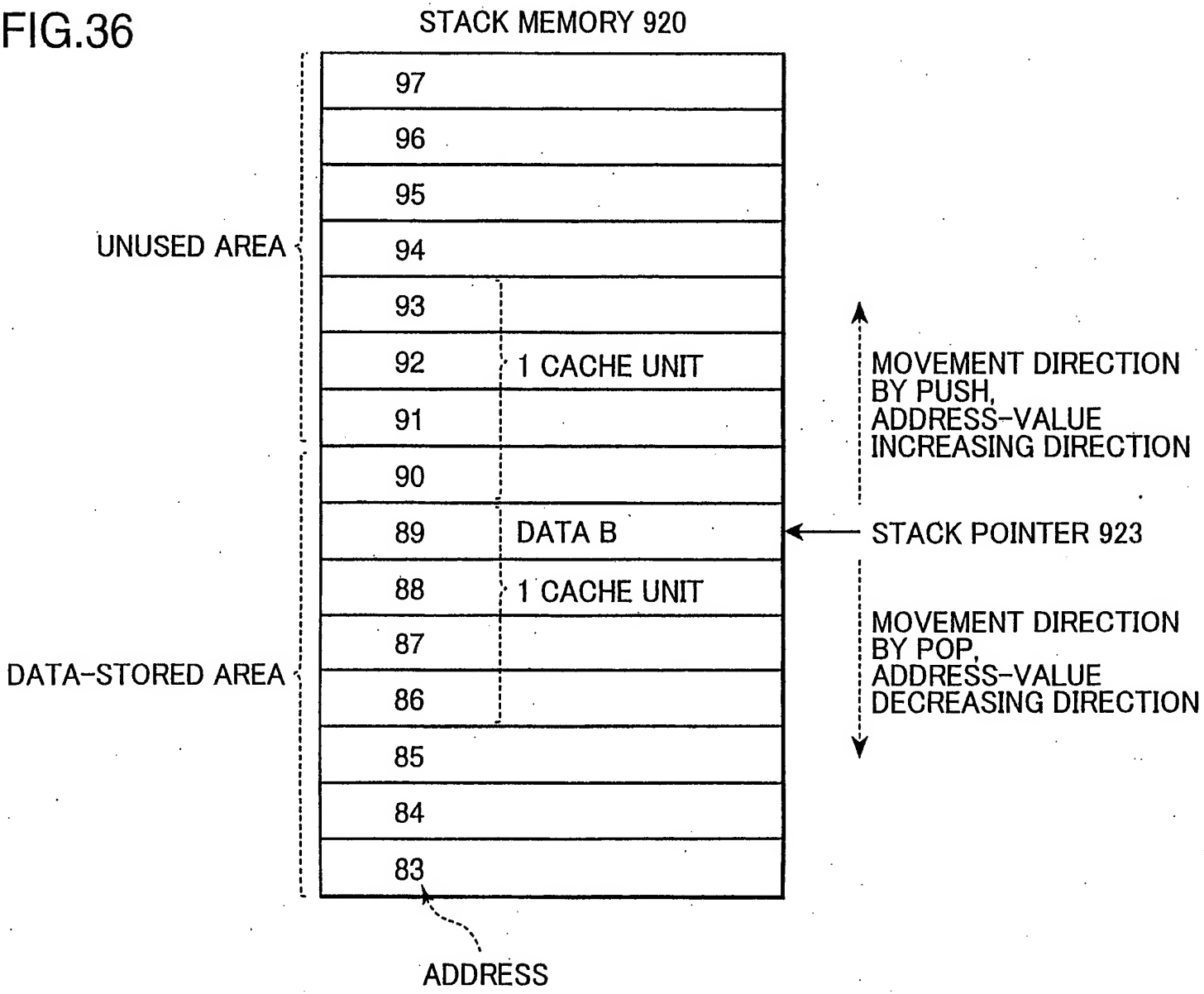

STACK MEMORY 920
97
96
95
94
93
92
91
90
89
88
87
86
85
84
83
UNUSED AREA
DATA-STORED AREA
1 CACHE UNIT
1 CACHE UNIT
DATA B
STACK POINTER 923
MOVEMENT DIRECTION BY PUSH, ADDRESS-VALUE INCREASING DIRECTION
MOVEMENT DIRECTION BY POP, ADDRESS-VALUE DECREASING DIRECTION
ADDRESS

FIG.37

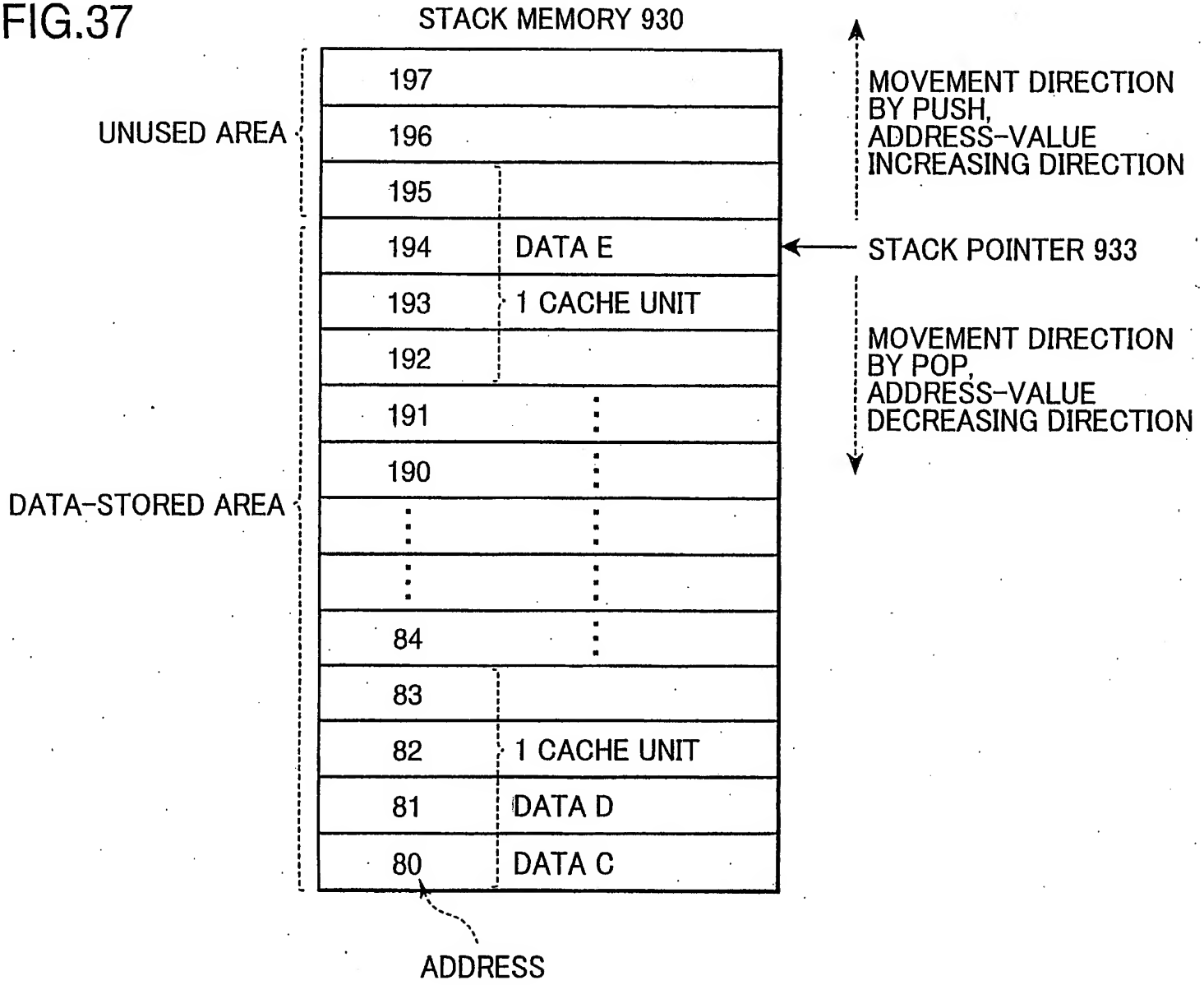
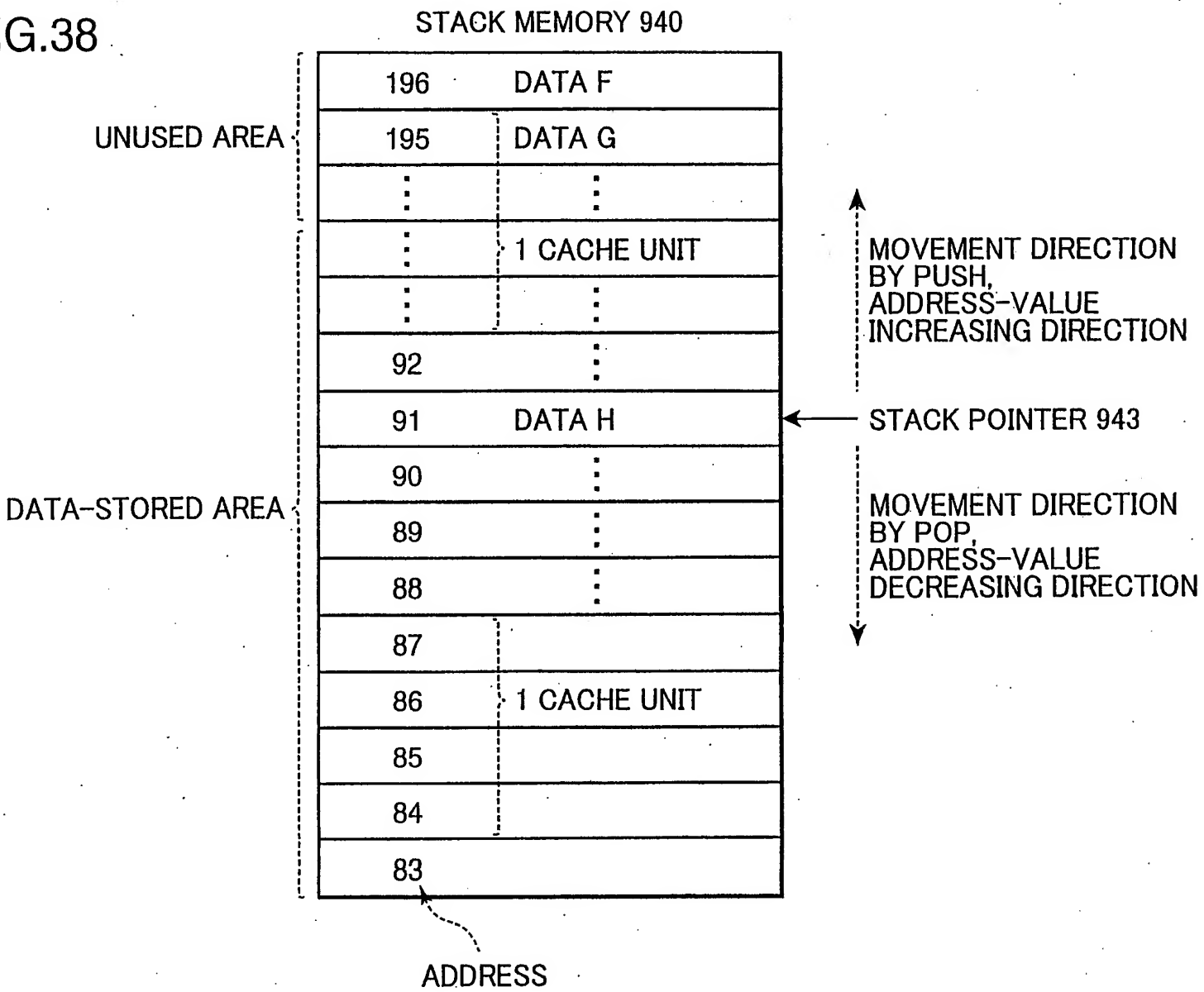

STACK MEMORY 930
UNUSED AREA
DATA-STORED AREA
197
196
195
194
193
192
191
190
84
83
82
81
80
DATA E
1 CACHE UNIT
1 CACHE UNIT
DATA D
DATA C
STACK POINTER 933
MOVEMENT DIRECTION BY PUSH, ADDRESS-VALUE INCREASING DIRECTION
MOVEMENT DIRECTION BY POP, ADDRESS-VALUE DECREASING DIRECTION
ADDRESS

FIG.38


STACK MEMORY 940
196
DATA F
UNUSED AREA
195
DATA G
1 CACHE UNIT
92
91
DATA H
STACK POINTER 943
90
DATA-STORED AREA
89
88
87
86
1 CACHE UNIT
85
84
83
ADDRESS
MOVEMENT DIRECTION BY PUSH, ADDRESS-VALUE INCREASING DIRECTION
MOVEMENT DIRECTION BY POP, ADDRESS-VALUE DECREASING DIRECTION